

Practical Electronics

The UK's premier electronics and computing maker magazine

Circuit Surgery

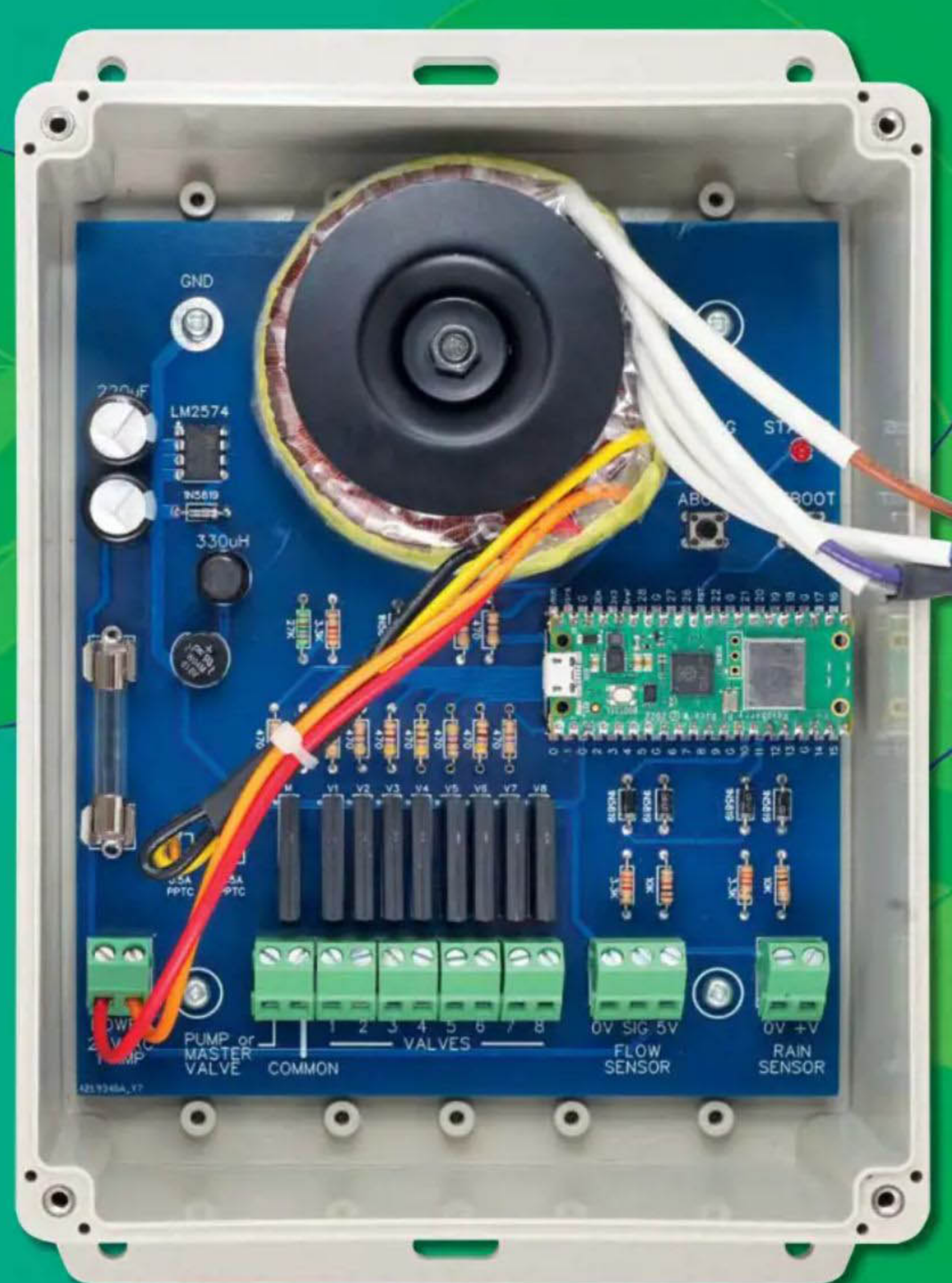
Topics in digital signal processing
– the frequency domain

Teach-In 2024

Create a Wi-Fi Access Point with
the ESP32 SoC microcontroller

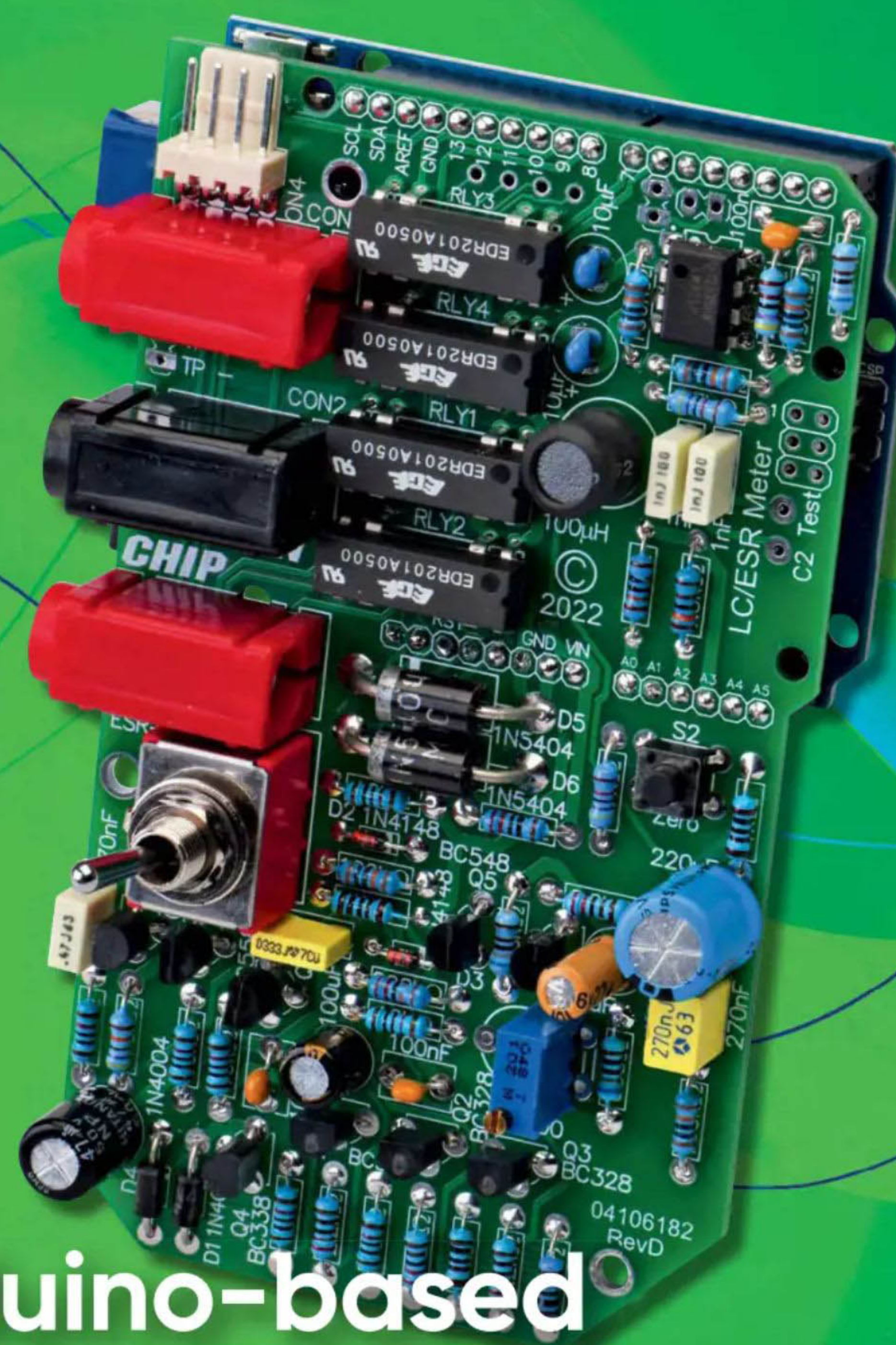
Audio Out

Mechanical
switches in audio



WebMite Watering System Controller

WIN!
Microchip
PolarFire SoC
Discovery Kit
WIN!



Arduino-based L/C+ESR Meter



Teach-In 2024 Learn to use the ESP32

WebMite: PicoMite meets the Web



PLUS!

Techno Talk – That makes so much sense!

Cool Beans – DIY logic probes

Net Work – Passkeys and online security

Aug 2024 £5.99





Meet New Cybersecurity Mandates Easily: PIC32CK SG MCU

Arm® Cortex®-M33 Core With HSM and TrustZone® Technology

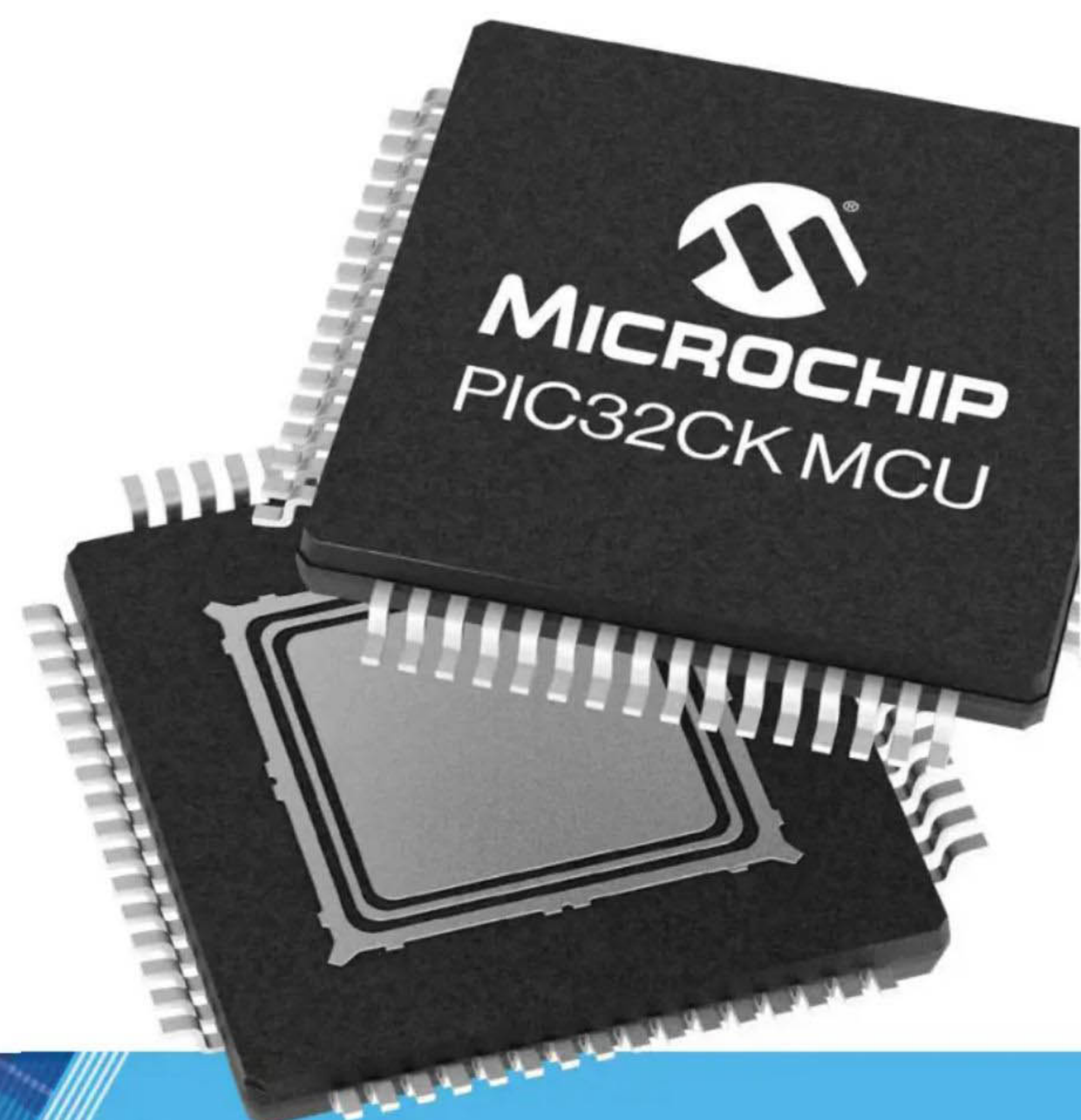
In anticipation of stringent cybersecurity standards, our PIC32CK SG MCU is your solution for addressing the pressing cybersecurity challenges without compromising performance.

The PIC32CK SG MCU family combines the security of a fully isolated Hardware Security Module with Arm's TrustZone® secure partitioning. This provides defense in depth against cyber-attacks for the most demanding applications.

Designed as a mid-range MCU, the PIC32CK SG MCU integrates advanced hardware security subsystems to surpass regulatory requirements effortlessly. With a high level of cyber protection integrated, designers can ensure their products meet and exceed cybersecurity mandates without sacrificing efficiency or speed.

Key Features

- Based on Cortex-M33 at 120 MHz
- 2 MB Dual Panel Flash/512 KB RAM for complex system
- 10/100 Ethernet, USB, CAN-FD, I2S, I2C
- Advanced key management
- Certificate authority
- Secure boot

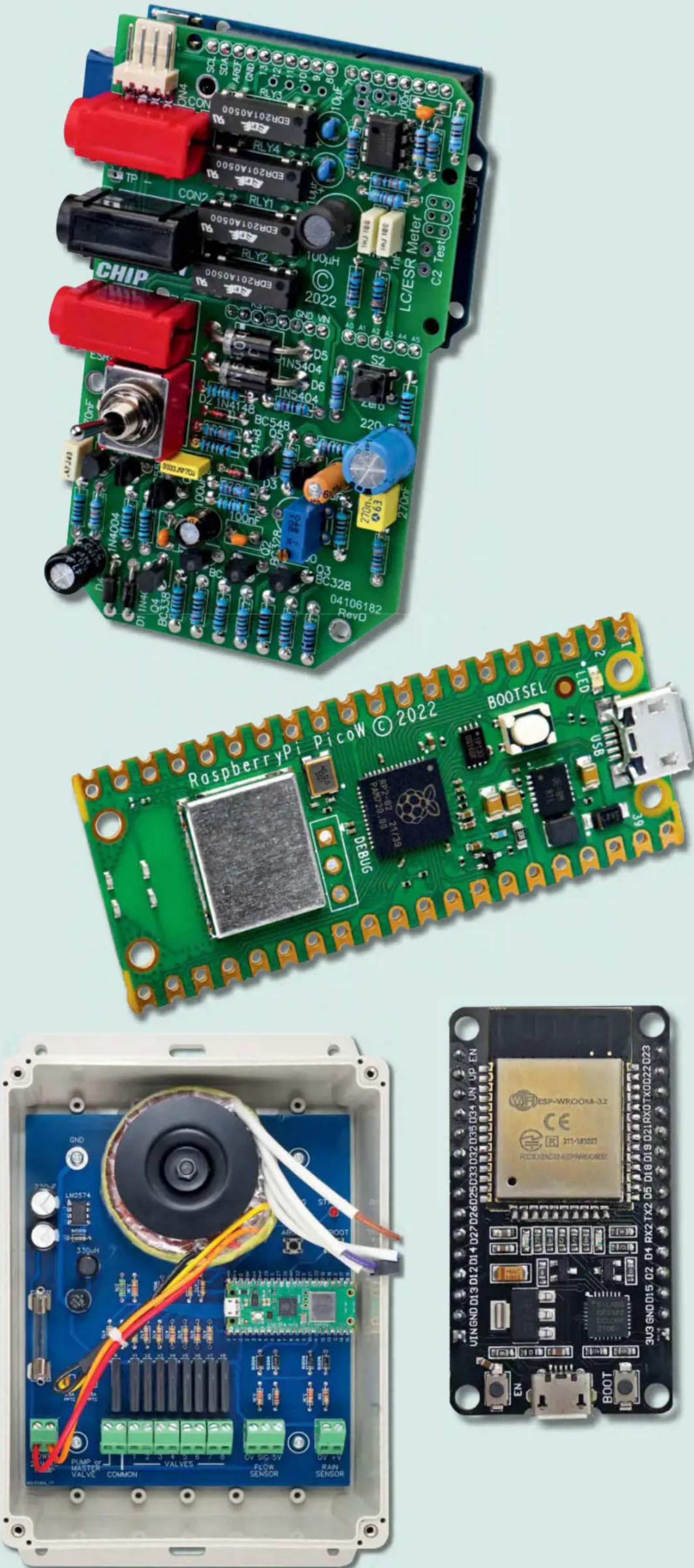


MICROCHIP



microchip.com/PIC32CKSGMCU

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2022 Microchip Technology Inc. All rights reserved.
MEC2572A-UK-06-24



Made in the UK.
Written in Britain, Australia,
the US and Ireland.
Read everywhere.

© Electron Publishing Limited 2024
Copyright in all drawings, photographs, articles,
technical designs, software and intellectual property
published in *Practical Electronics* is fully protected,
and reproduction or imitation in whole or in part are
expressly forbidden.
The September 2024 issue of *Practical Electronics* will be
published on Thursday, 1 August 2024 – see page 72.

Projects and Circuits

LC and ESR Meter by Steve Matthysen	16
This enhancement to our Wide-Range Digital LC Meter (June 2019) adds capacitor ESR measurement – useful for diagnosing faulty equipment because increasing ESR over time is one of the most common ways electrolytic capacitors fail.	
WebMite by Geoff Graham with firmware by Peter Mather	28
What happens when the PicoMite meets the web? Find out with this powerful project that combines a Raspberry Pi Pico with MMBasic, Wi-Fi and Internet connectivity. We put it to good use in this month's Watering System Controller	
WebMite-based Watering System Controller by Geoff Graham	34
This advanced Watering System Controller does it all. It can change the watering schedule depending on the seasons, check the weather forecast before watering and alert you to a burst pipe or blocked sprinkler. Best of all, it's easy to set up and use.	

Series, Features and Columns

Techno Talk by Max The Magnificent	8
That makes so much sense!	
Net Work by Alan Winstanley	10
Reports on passkeys to improve online security and eliminate password logins, plus the transfer of data to a new phone – some hints and tips.	
Circuit Surgery by Ian Bell	44
Topics in digital signal processing – the frequency domain	
Audio Out by Jake Rothman	48
Switching in audio equipment – Part 3	
Max's Cool Beans by Max The Magnificent	52
Arduino Bootcamp – Part 20	
Teach-In 2024 – Learn electronics with the ESP32 by Mike Tooley	60
Part 6 – the ESP32 as a Wi-Fi Access Point	

Regulars and Services

Wireless for the Warrior	2
Subscribe to Practical Electronics and save money	4
NEW! Practical Electronics back issues DOWNLOADS – 2022 now available!	6
Reader services – Editorial and Advertising Departments	7
Editorial	7
Exciting news!	
Exclusive Microchip reader offer	9
Win a Microchip PolarFire SoC Discovery Kit	
PE Teach-In 9	27
Teach-In bundle – what a bargain!	33
PE Teach-In 8	43
Classified ads and Advertiser index	69
Practical Electronics PCB Service	70
PCBs for Practical Electronics projects	
Next month! – highlights of our next issue of Practical Electronics	72

WIRELESS FOR THE WARRIOR

by LOUIS MEULSTEE

THE DEFINITIVE TECHNICAL HISTORY OF RADIO COMMUNICATION EQUIPMENT IN THE BRITISH ARMY

The *Wireless for the Warrior* books are a source of reference for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

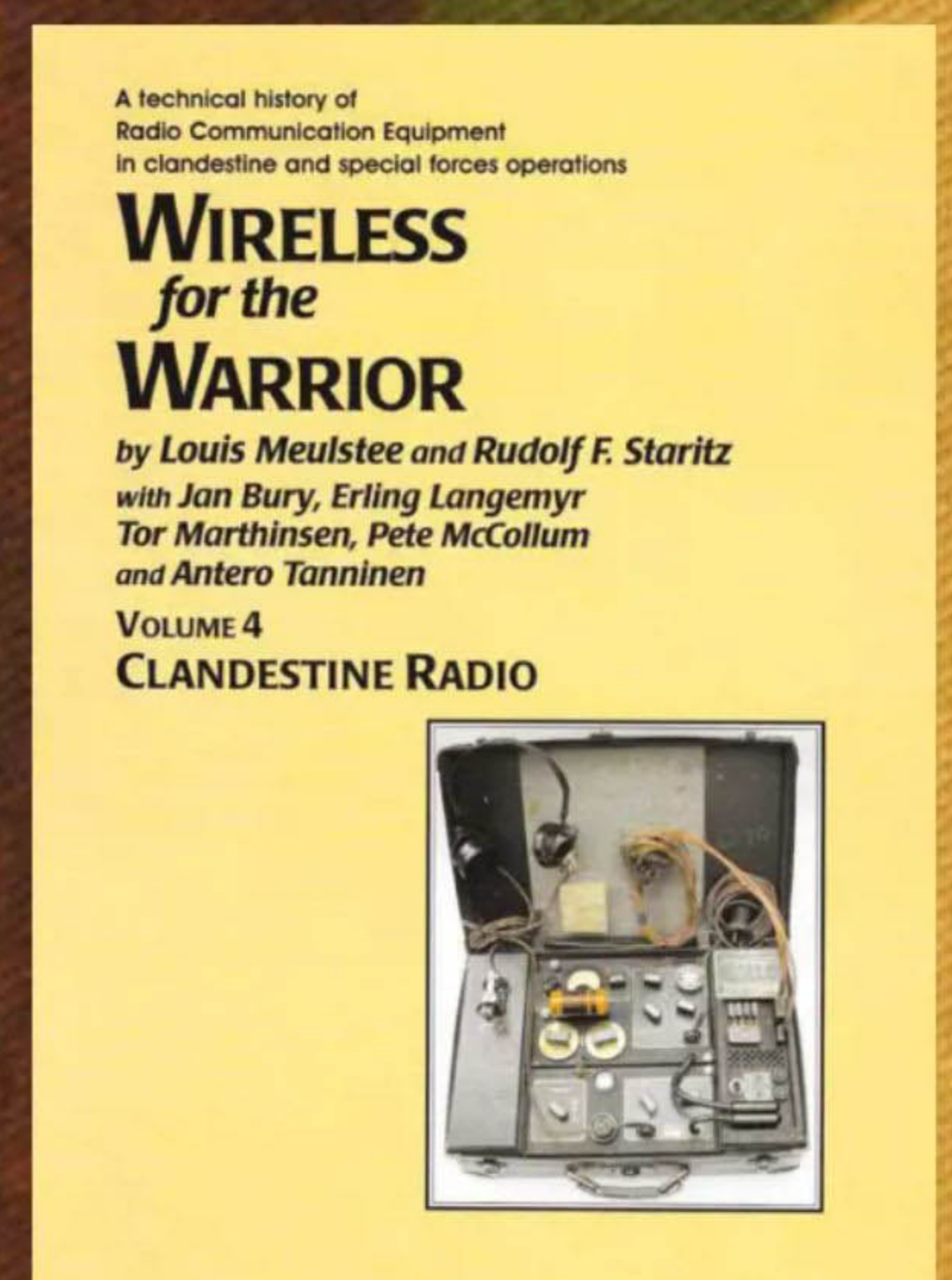
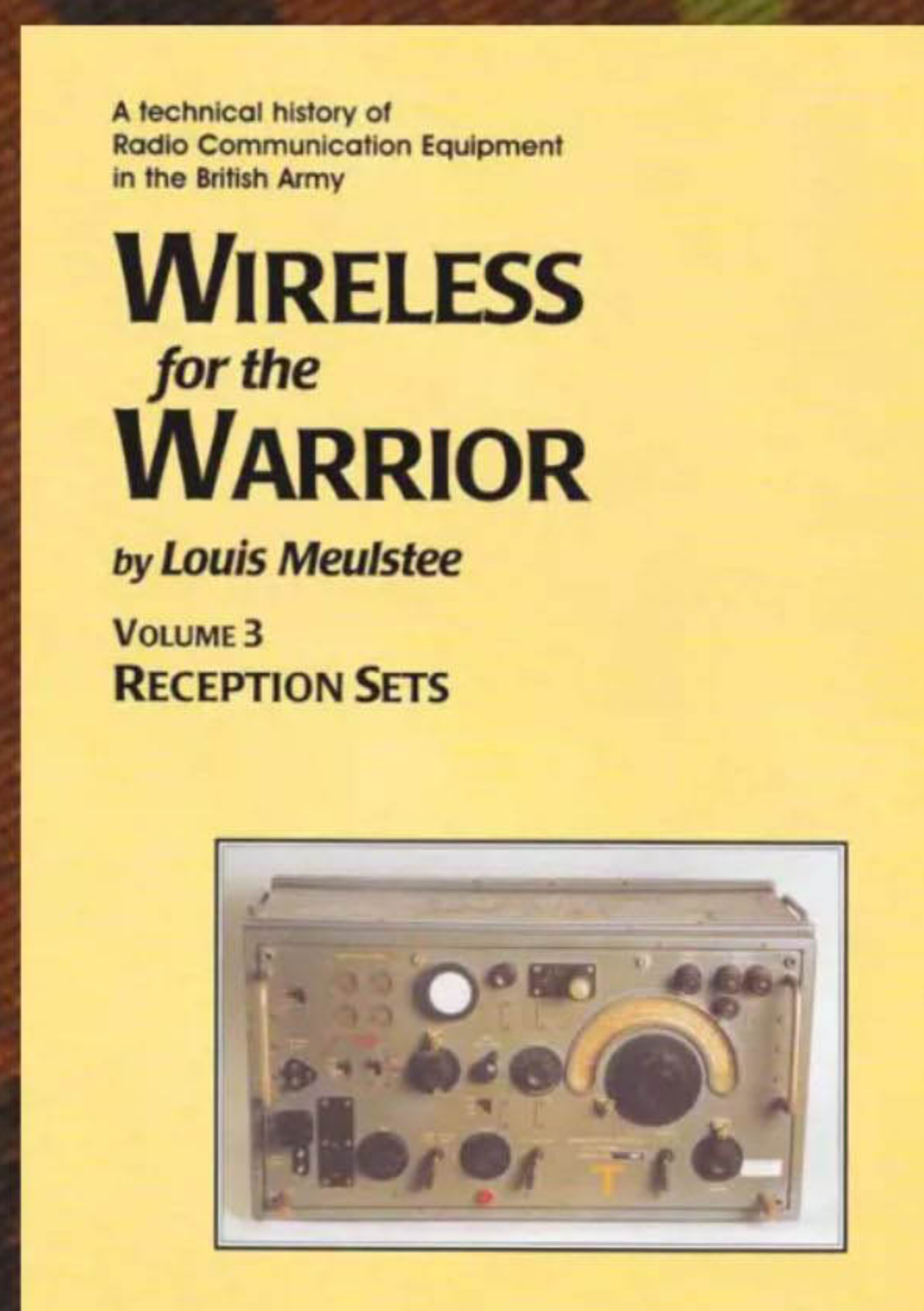
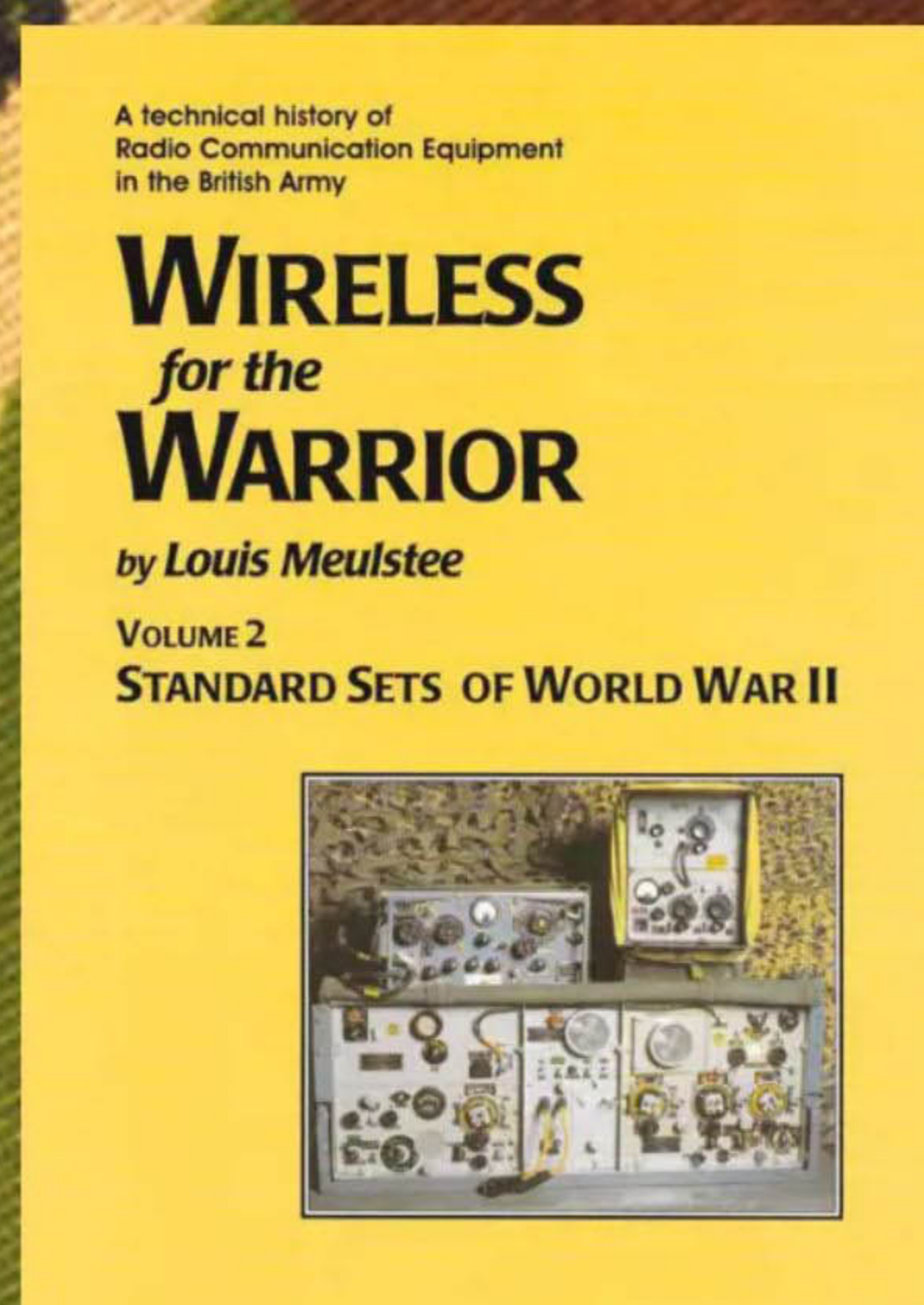
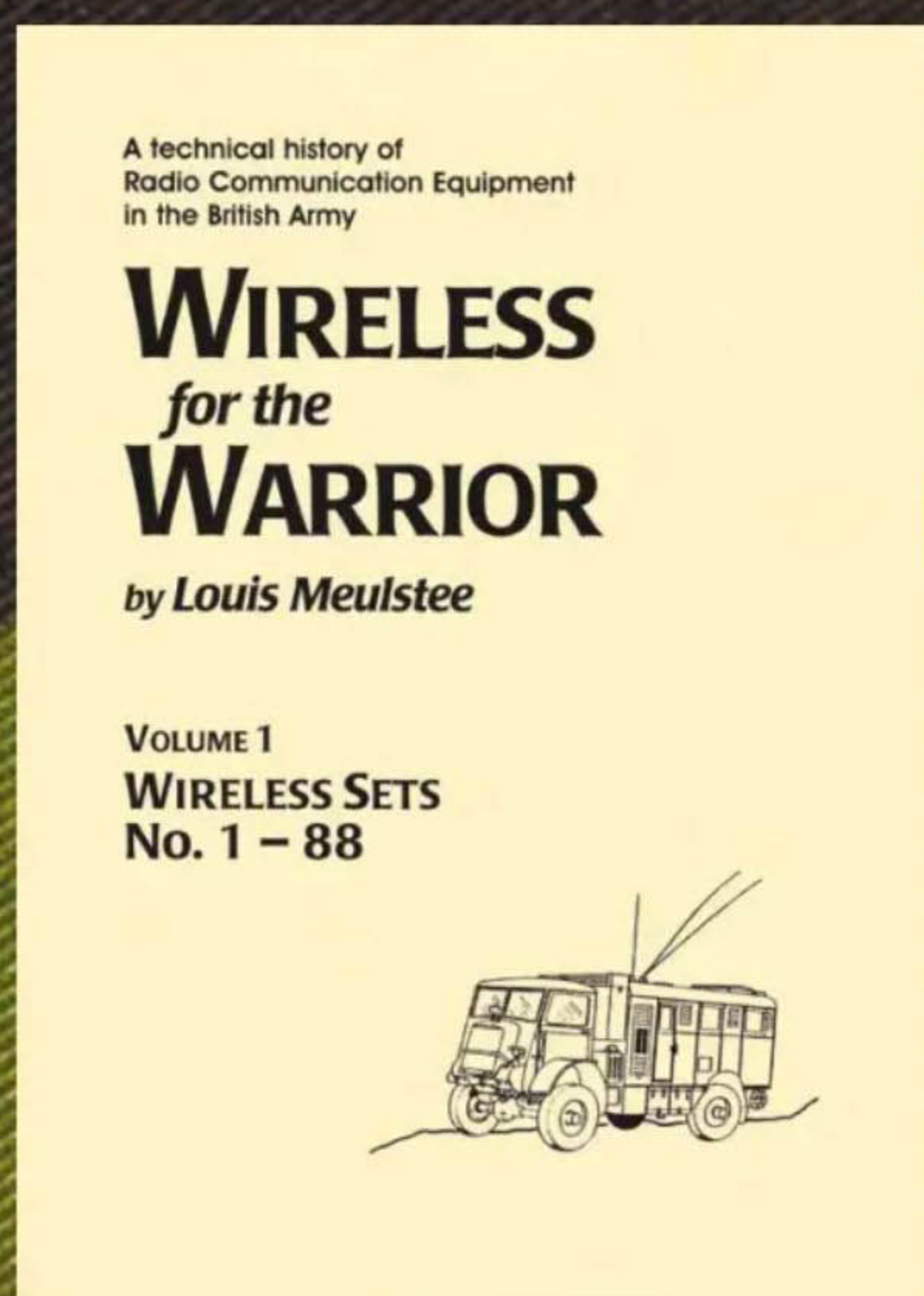
The books are very detailed and include circuit diagrams, technical specifications and alignment data, technical development history, complete station lists and vehicle fitting instructions.

Volume 1 and Volume 2 cover transmitters and transceivers used between 1932-1948. An era that starts with positive steps taken to formulate and develop a new series of wireless sets that offered great improvements over obsolete World War I pattern equipment. The other end of this timeframe saw

the introduction of VHF FM and hermetically sealed equipment.

Volume 3 covers army receivers from 1932 to the late 1960s. The book not only describes receivers specifically designed for the British Army, but also the Royal Navy and RAF. Also covered: special receivers, direction finding receivers, Canadian and Australian Army receivers, commercial receivers adopted by the Army, and Army Welfare broadcast receivers.

Volume 4 covers clandestine, agent or 'spy' radio equipment, sets which were used by special forces, partisans, resistance, 'stay behind' organisations, Australian Coast Watchers and the diplomatic service. Plus, selected associated power sources, RDF and intercept receivers, bugs and radar beacons.



ORDER YOURS TODAY!

JUST CALL 01202 880299 OR VISIT www.electronpublishing.com



Quasar Electronics Limited
PO Box 6935, Bishops Cleeve
CM23 4WP, United Kingdom

Tel: 01279 467799
E-mail: sales@quasarelectronics.co.uk
Web: quasarelectronics.co.uk

All prices include 20% VAT. Free UK mainland delivery on orders over £60.
Postage & Packing Options (Up to 1kg gross weight): UK Standard 2-5 Day Delivery - £4.95 : UK Mainland Next Day Delivery - £9.95 : Please order online if you reside outside the UK (our website will calculate postage for you).
Payment: We accept all major credit/debit cards. Make UK cheques/PO's payable to Quasar Electronics Limited and include P&P detailed above.
!! Order online for reduced price postage and fast despatch !!
Please visit our online shop now for full details of over 1000 electronic kits, projects, modules and publications. Discounts for bulk quantities.



 **QuasarElectronics.co.uk**

Solutions for Home, Education & Industry Since 1993



Great Brands - Official Main Dealer

Electronic Kits & Modules

We have a massive selection of self-assembly electronic kits and pre-assembled modules. Please see the full range on our website or call for details.

LED Buddy / LED Tester Kit

Hold any type of LED to the contact pads to see its polarity, forward voltage & the recommended series resistor value. Adjustable target current & forward voltage. Great design aid.
Order Code: WSM198 - £16.66



12-in-1 Solar Hydraulic Construction Kit

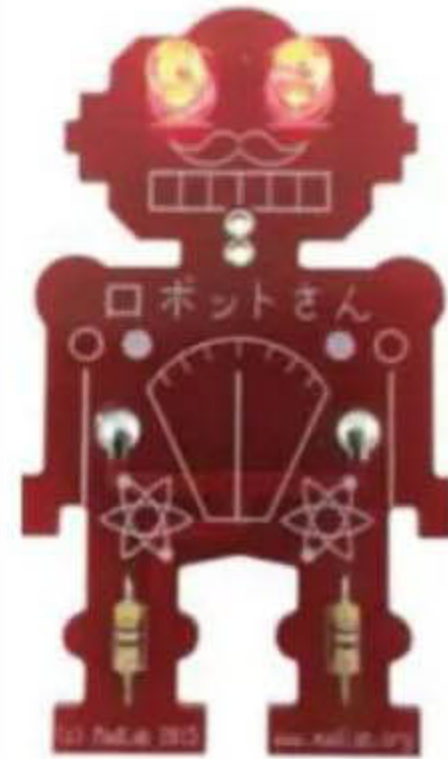
Solar & hydraulic powered robot can be transformed into twelve different animals and mechanical robots (monkey, T-Rex, scorpion, excavator, etc). Moves easily. Provides great interaction with kids. Teaches the benefit of alternative energy. Aged 14+.
Order Code: KSR17 - £29.95.



Stereo Ultrasonic Bat Detector Kit



Educational & fun kit converts high frequency sounds (20 - 90kHz) normally imperceptible to humans like bat signals into audible noise. Can also help detect failures in machines, engines, etc. Stereo feature adds the possibility to pinpoint the source. Frequency range 20-90kHz. 3.5mm jack output. 129x60x40mm. Requires 3x AA batteries and stereo headphones (not included).
Order Code: WSAK8118 - £20.39



NEW! Mr Robot Kit

A fun, wearable electronic gadget that you can pin to your clothes. Easy to solder beginner's kit. Two flashing bicolour LEDs resemble the eyes. 65 x 40 x 15 mm. CR2032 battery powered (not included).
Order Code: WSL108 - 6.68

LED Electronic Dice Kit

Still popular! A must-build soldering kit for all beginners. Dice slowly rolls to stop on a random number when the push button is released. 9V battery (not included). Ideal for educational courses. Wide selection of component types.
Order Code: MK109 - £7.08



3 Channel RGB LED Light Organ Kit



3 outputs react to different sound frequencies. On-board microphone picks up surrounding sound or music and drives the low, mid & high frequency outputs. Connect 12-24Vdc RGB or separate colour LED strips or LEDs (not included). Master & separate channel sensitivity adjustment. 1.25 A max. per channel. Panel mountable fascia 127 x 44mm. Terminal block or jack power supply connection. Control knobs included.
Order Code: WSL209 - £14.10



Signal Generator Kit

Sine wave, triangle, square wave and integrator (selectable through jumper) with 0 - 100Vrms adjustable output level. 1kHz (approx.) fixed signal frequency. 9V battery (not included).
Order Code: WSAH105 - £6.12

Bidirectional DC Motor Speed Controller

Control the speed of most common DC motors (rated up to 28Vdc/5A) from fully OFF to fully ON in both directions. Single potentiometer controls speed & direction. Screw terminal block connectors. PCB: 90x42mm. **Not suitable for use with lead acid batteries!**
Kit Order Code: 3166KT - £19.96
Assembled Order Code: AS3166 - £29.95

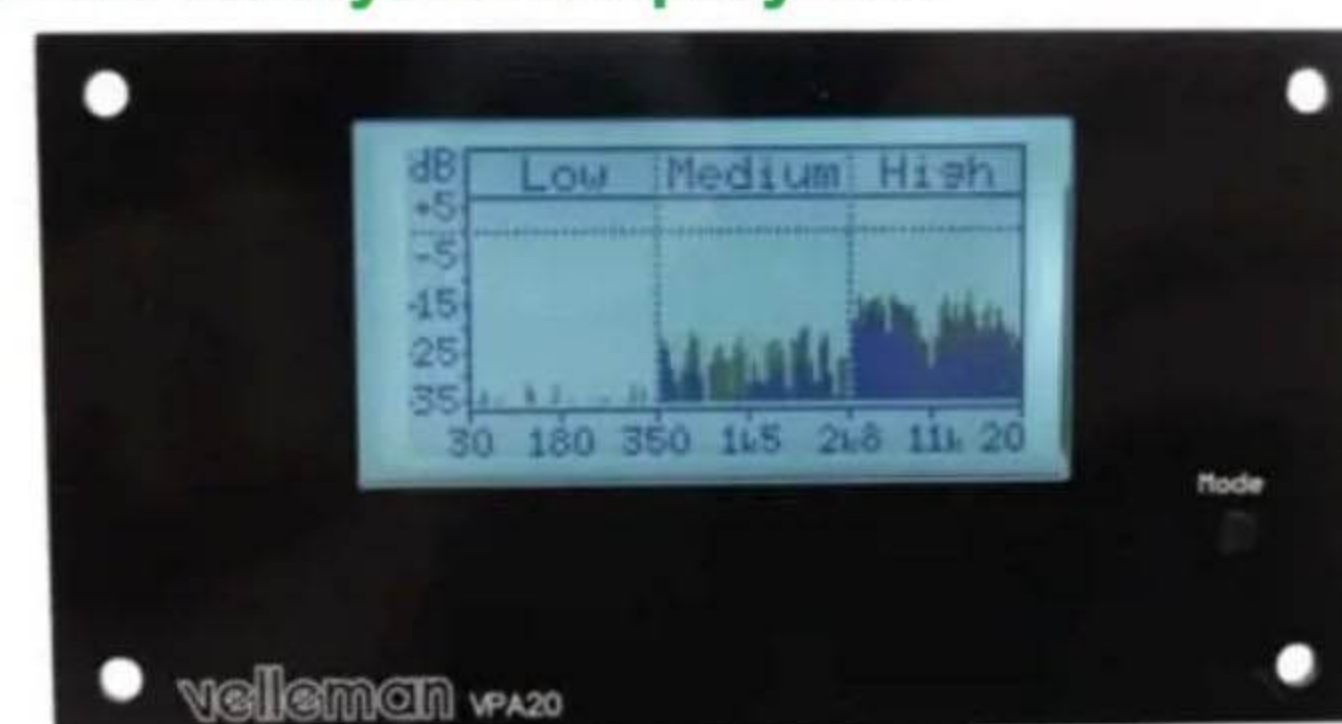


Digitally Controlled FM Radio Kit

Build your own modern, high quality FM receiver project with excellent sensitivity powered by a simple 9V PP3 battery (not included). Auto-seeking button. 4 station presets. Volume control. Excellent learning project for schools and colleges.
Order Code: WSAH194 - £20.39



Audio Analyser Display Kit



Small, compact LCD display, ideal for panel mounting. Give your homemade audio gear a high-tech look. Upgrade existing equipment. Provides Peak Power, RMS Power, Mean dB, Peak dB, Linear Audio Spectrum And 1/3 Octave Audio Spectrum. Auto / Manual range selection. Peak-hold function. Speaker impedance selection.
Order Code: K8098 - £38.39

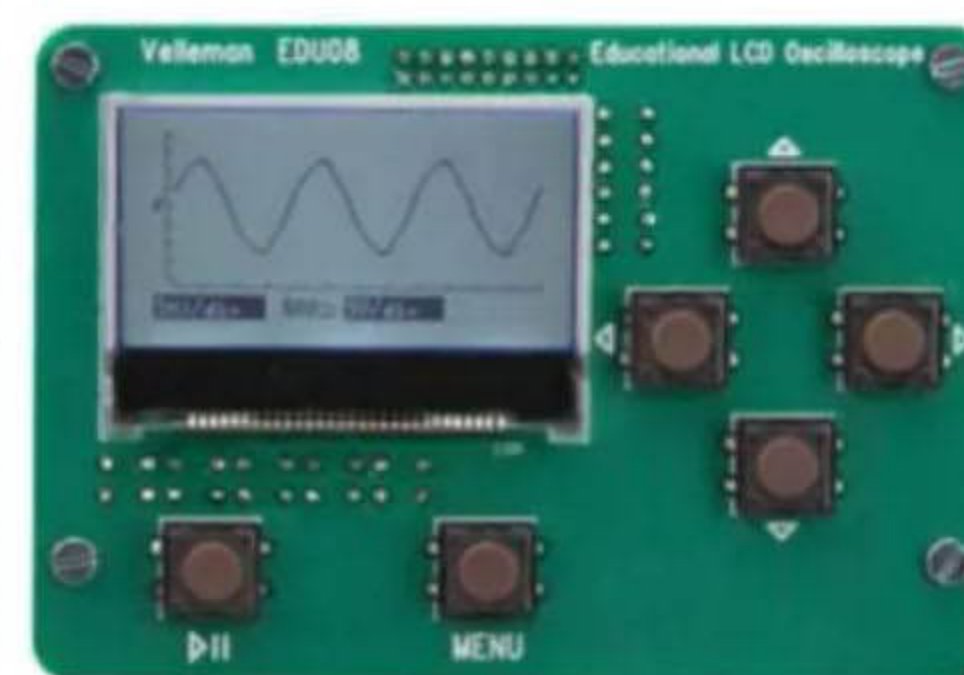
Electronic Component Tester Kit

Build your own versatile component tester. Shows value and pin layout information for resistors (0.1 Ohm resolution, max. 50M Ohm), coils (0.01mH - 20H), capacitors (28p - 100mF), diodes, BJT, JFET, E-IGBT, D-IGBT, E-MOS & D-MOS.
Order Code: WSM18115 - £44.15

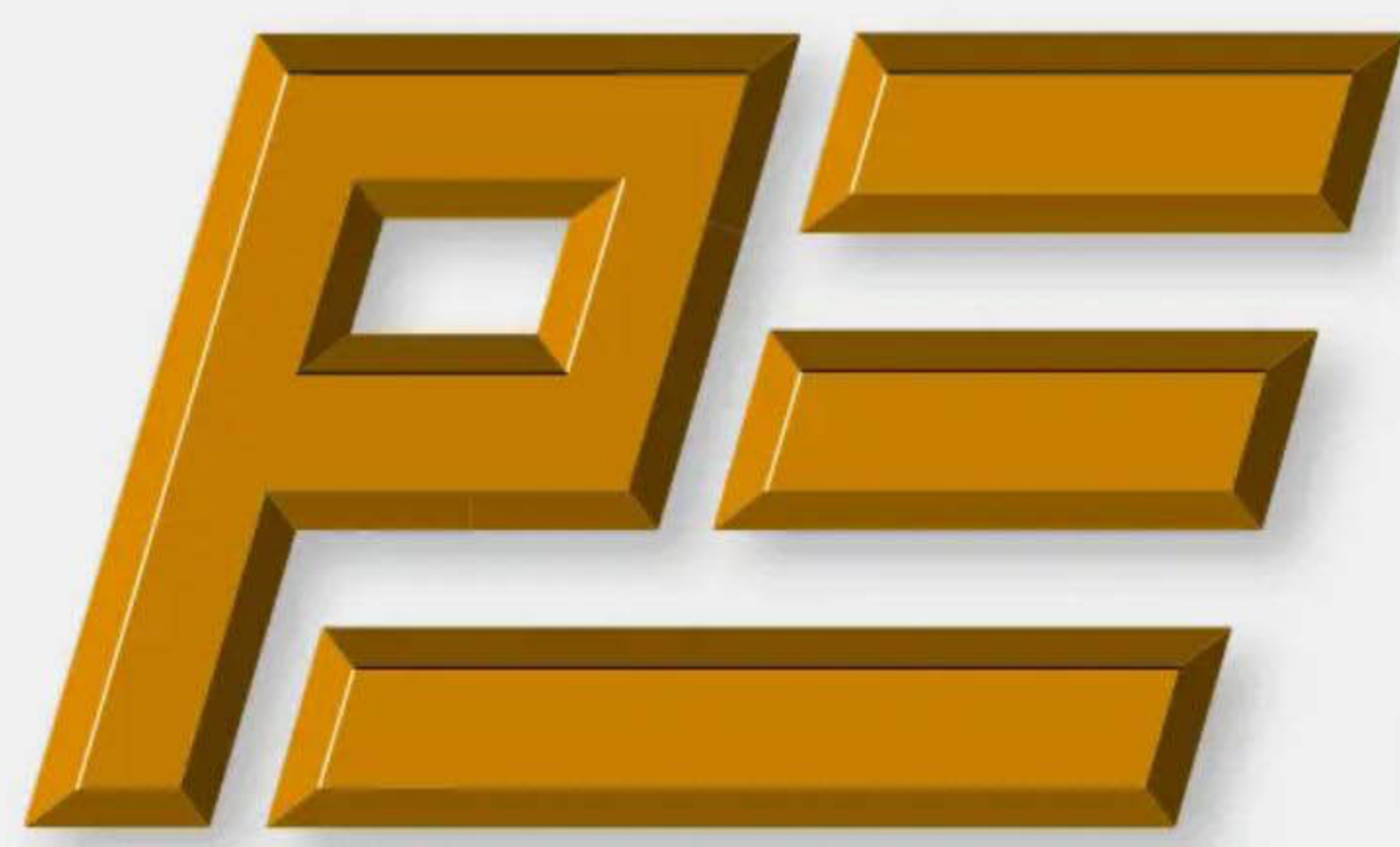


LCD Oscilloscope Educational Kit

Build your own LCD oscilloscope with this exciting new kit. Learn how to read signals. See the electronic signals you learn about displayed on your own LCD oscilloscope. Despite the low cost, this oscilloscope kit has a lot of features found on expensive units like signal markers, frequency, dB, true RMS readouts and more. A powerful auto-setup function will get you going in a flash!
Order Code: WSEDU08 - £48.54



Card Sales & Enquiries | **01279 467799**



Practical Electronics

UK readers **SAVE £1** on every issue

SUBSCRIBE NOW!

01202 087631



Take out a one-year subscription and save more than **£10** over the year.

Even better – save over **£1** per issue if you subscribe for two years – a total saving of **£26**.

Overseas rates represent exceptional value

- You also:
- Avoid any cover price increase for the duration of your subscription
 - Get your magazine delivered to your door each month
 - Guarantee a copy, even if your local newsagent sells out

- Order by:
- Phone or post with a cheque, postal order or credit card
 - Or order online at: www.electronpublishing.com



Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £33.99, 12-months £59.99, 24-months £114.99

Europe Airmail: 6-months £38.99, 12-months £69.99, 24-months £129.99

Rest Of The World Airmail: 6-months £46.99, 12-months £84.99, 24-months £164.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631

Email: pesubs@selectps.com

Also via our online shop at: www.electronpublishing.com

Subscriptions start with the next available issue. We accept MasterCard or Visa.

(For past issues see the Back Issues page.)



DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £29.99 for 12 months, more details at: www.electronpublishing.com

SUBSCRIPTION ORDER FORM

- ☐ 6 Months: UK £33.99, Europe £38.99 (Airmail), Rest of the World £46.99 (Airmail)
- ☐ 1 Year: UK £59.99, Europe £69.99 (Airmail), Rest of the World £84.99 (Airmail)
- ☐ 2 Years: UK £114.99, Europe £129.99 (Airmail), Rest of the World £164.99 (Airmail)

To: Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631

Email: pesubs@selectps.com

I enclose payment of £
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard



My card number is:
Please print clearly, and check that you have the number correct

Card Security Code Valid From Date.....
(The last 3 digits on or just under the signature strip)

Card Ex. Date

Name

Address

Post code Tel.

Email

Subscriptions start with the next available issue.

Subscribe to **SILICON CHIP**

Australia's top electronics magazine

SILICON CHIP is one of the best DIY electronics magazines in the world. Each month is filled with a variety of projects that you can build yourself, along with features on a wide range of topics from in-depth electronics articles to general tech overviews.

	Print	Combined	Online
6 months	\$100	\$110	\$50
	≈£53	≈£59	≈£27
	≈€62	≈€68	≈€31
1 year	\$195	\$215	\$95
	≈£105	≈£115	≈£51
	≈€120	≈€132	≈€58
2 years	\$380	\$415	\$185
	≈£205	≈£223	≈£100
	≈€234	≈€255	≈€114

- Combined subscriptions include both the printed magazine and online access.
- Prices are in Australian dollars (AUD) unless otherwise specified.
- Prices listed are for everywhere in the world except Australia and New Zealand. View siliconchip.com.au/Shop/SubRates for a full list of current prices.
- Price estimates are just that and should only be used as a guide, we primarily take payment in the form of Australian dollars.

Try our Online Subscription – now with PDF downloads!

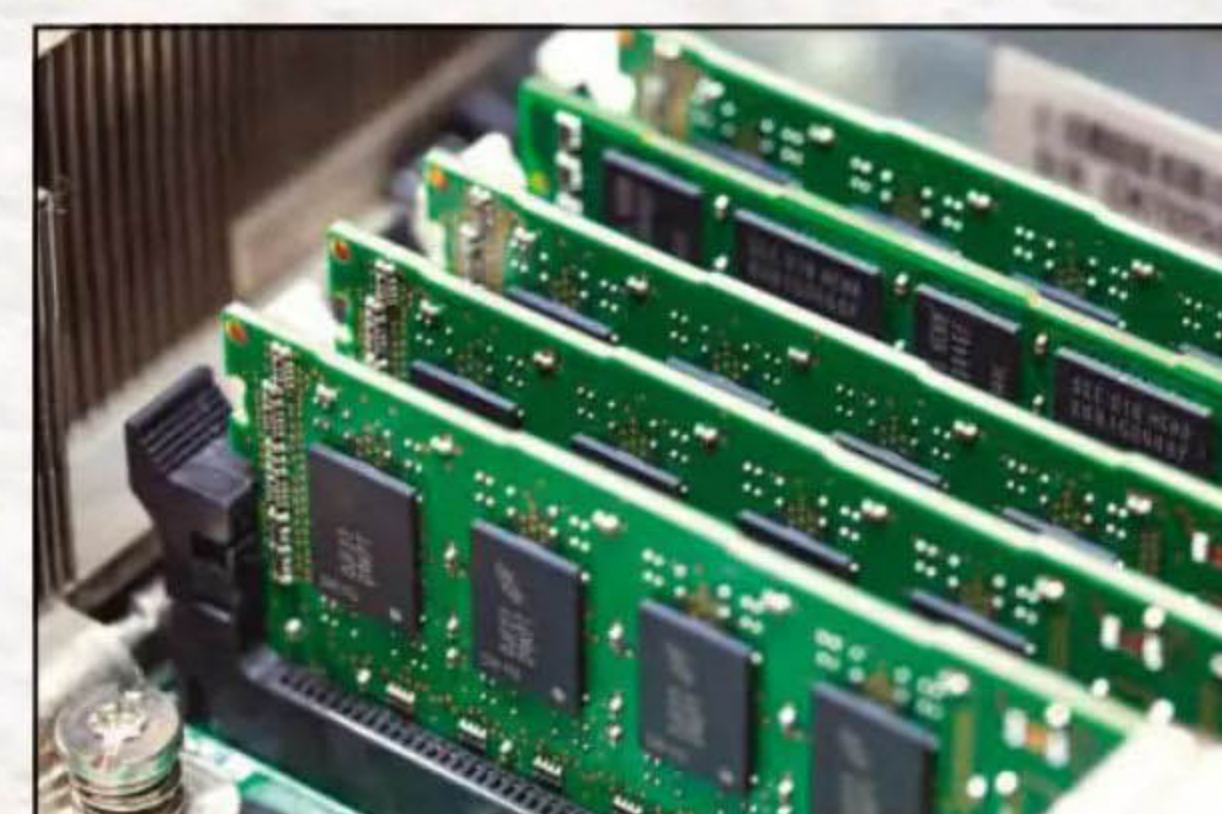
An online issue is perfect for those who don't want too much clutter around the house and is the same price worldwide.

Issues can be viewed online, or downloaded as a PDF.

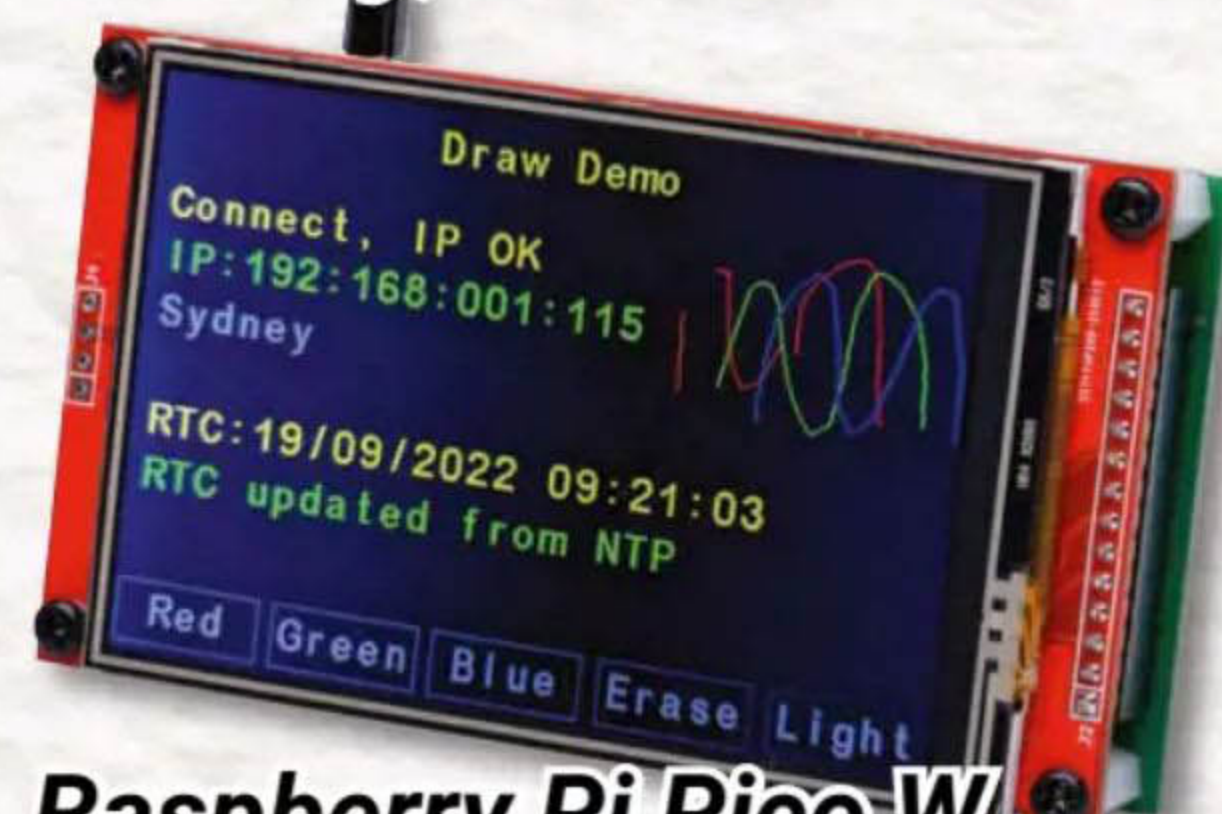
To start your subscription go to
siliconchip.com.au/Shop/Subscribe



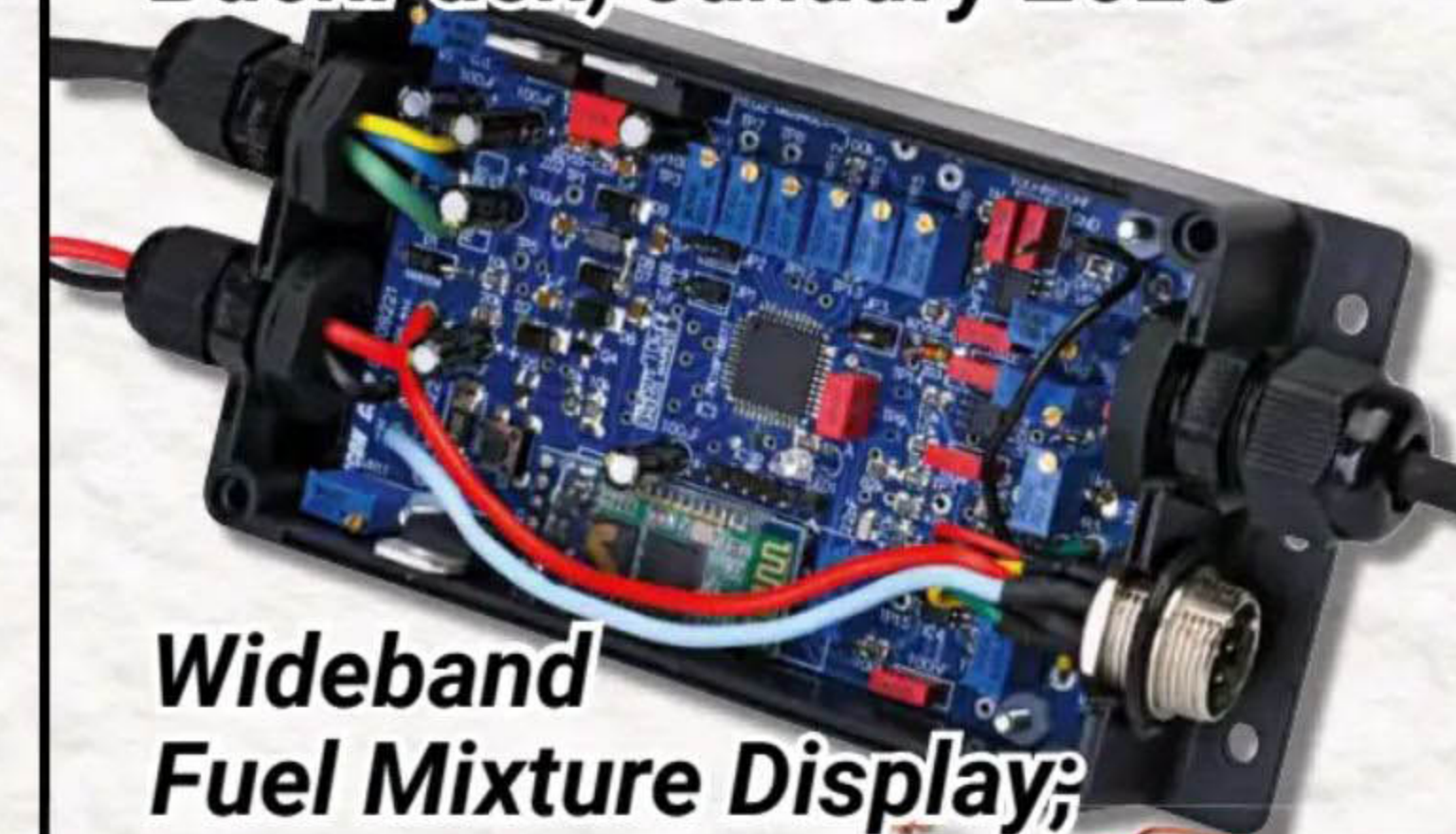
**Published in
SILICON CHIP**



**The History of Computer
Memory; Jan-Feb 2023**



**Raspberry Pi Pico W
BackPack; January 2023**



**Wideband
Fuel Mixture Display;
April to June 2023**



**Silicon Chirp;
April 2023**



**Advanced SMD
Test Tweezers;
February & March 2023**

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!
From just £5.95 each

VOL 28 – January to December 1999 – **£5.95**
VOL 29 – January to December 2000 – **£6.95**
VOL 30 – January to December 2001 – **£7.95**
VOL 31 – January to December 2002 – **£7.95**
VOL 32 – January to December 2003 – **£7.95**
VOL 33 – January to December 2004 – **£7.95**
VOL 34 – January to December 2005 – **£8.95**
VOL 35 – January to December 2006 – **£8.95**
VOL 36 – January to December 2007 – **£8.95**
VOL 37 – January to December 2008 – **£8.95**
VOL 38 – January to December 2009 – **£10.95**
VOL 39 – January to December 2010 – **£11.95**

VOL 40 – January to December 2011 – **£12.95**
VOL 41 – January to December 2012 – **£14.95**
VOL 42 – January to December 2013 – **£15.95**
VOL 43 – January to December 2014 – **£16.95**
VOL 44 – January to December 2015 – **£17.95**
VOL 45 – January to December 2016 – **£19.95**
VOL 46 – January to December 2017 – **£22.95**
VOL 47 – January to December 2018 – **£27.95**
VOL 48 – January to December 2019 – **£27.95**
VOL 49 – January to December 2020 – **£29.95**
VOL 50 – January to December 2021 – **£32.95**
VOL 51 – January to December 2022 – **£34.95 – NEW!**

FIVE-YEAR collections – 60 issues!
From just £14.95 each

PE2832: Jan 99 to Dec 03 – **£14.95**
PE2933: Jan 00 to Dec 04 – **£14.95**
PE3034: Jan 01 to Dec 05 – **£14.95**
PE3135: Jan 02 to Dec 06 – **£14.95**
PE3236: Jan 03 to Dec 07 – **£14.95**
PE3337: Jan 04 to Dec 08 – **£14.95**
PE3438: Jan 05 to Dec 09 – **£19.95**
PE3539: Jan 06 to Dec 10 – **£25.95**
PE3640: Jan 07 to Dec 11 – **£25.95**
PE3741: Jan 08 to Dec 12 – **£25.95**

PE3842: Jan 09 to Dec 13 – **£33.95**
PE3943: Jan 10 to Dec 14 – **£33.95**
PE4044: Jan 11 to Dec 15 – **£33.95**
PE4145: Jan 12 to Dec 16 – **£35.95**
PE4246: Jan 13 to Dec 17 – **£37.95**
PE4347: Jan 14 to Dec 18 – **£38.95**
PE4448: Jan 15 to Dec 19 – **£39.95**
PE4549: Jan 16 to Dec 20 – **£42.95**
PE4650: Jan 17 to Dec 21 – **£44.95 – NEW!**

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£59.95

PE2832: Jan 99 to Dec 03
PLUS!

PE3337: Jan 04 to Dec 08
PLUS!

PE3842: Jan 09 to Dec 13
PLUS!

PE3447: Jan 14 to Dec 18

Do you have the 15-year bundle?

Upgrade to the new 20-year bundle for just £14.95 with: **PE15to20**

Download your collections at: www.electronpublishing.com

Editorial offices

Practical Electronics
Electron Publishing Limited
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 01273 777619
Mob 07973 518682
Fax 01202 843233
Email pe@electronpublishing.com
Web www.electronpublishing.com

Advertisement offices

Practical Electronics Adverts
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 01273 777619
Mob 07973 518682
Email pe@electronpublishing.com

Editor Matt Pulzer
General Manager Louisa Pulzer
Digital subscriptions Stewart Kearn **Tel** 01202 880299
Online Editor Alan Winstanley
Web Systems Kris Thain
Publisher Matt Pulzer

Print subscriptions

Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel 01202 087631
Email pesubs@selectps.com

Technical enquiries

We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We cannot provide data or answer queries on articles or projects that are more than five years old.

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits

All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies

We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements

Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment

We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Exciting News!

Change is in the air. *Practical Electronics* is the world's longest-running hobby electronics magazine. Our first issue came out in November 1964 when commercial transistors had been available for less than a decade and ICs were only just beginning to appear – at eye-watering prices that only NASA could afford. Many fine magazines have appeared (and sadly disappeared) in the last 60 years, but *PE* has endured and thrived. I'm sure that one of the main reasons *PE* has survived is the quality of its content – thanks to the very hard work of all who write and work for the magazine. But there is another reason – we've never been afraid to change, and now it's time for a new chapter in *PE's* long-running story. We are joining the *Silicon Chip* publishing stable.

What does this mean for Practical Electronics?

Silicon Chip is an Australian magazine with which we have shared content since the beginning of 2008. They have provided sophisticated projects of the highest quality and it's been a very successful relationship. So successful in fact, that when I mentioned to its editor/publisher Nicholas Vinen that I was thinking of retiring we decided to work out a way to ensure *PE* wouldn't retire with me.

So, the big question is where does *PE* go from here? The short answer is most things will stay exactly as they are. All of *PE's* authors will continue to contribute exactly as they do now. Nicholas is absolutely clear that *PE* must not become a '*Silicon Chip* light', that would be pointless. If anything, the magazine will get better – Nicholas intends to up the pagination which means more pages and more content, all of which will benefit you the reader.

Thank you

I first started working on *PE* nearly 30 years ago with a small job producing its circuit diagrams. In 2008 Mike Kenward, the editor/publisher asked me if I'd consider becoming *PE's* editor. I jumped at the opportunity, and it's been an exciting and rewarding role. Then, at the start of 2019 I became publisher, just in time to catch the full brunt of the pandemic, but we survived!

Throughout my time with *PE* I have had fantastic support from our many talented contributors – *PE* is nothing without its creative scribes – thank you one and all; in no particular order: Alan Winstanley, Jake Rothman, Ian Bell, Clive 'Max' Maxfield, Mike Tooley, Julian Edgar, Mike Hibbett, Phil Boyce, Mark Nelson, Barry Fox, Stewart Kearn, Kris Thain, the late John Becker, Mike Kenward... and to anyone I've left off this list, apologies! Last, but by no means least, thank you to all our many loyal readers and subscribers. You've kept us going and I'm sure your continued support will enable Nicholas to carry on bringing you the very best in electronics publishing over the coming years.

Matt Pulzer
Publisher

That makes so much sense!

Techno Talk

Max the Magnificent

I'm always amazed by the cunning creations of the pioneers of yesteryear, especially when I consider the rudimentary sensors they had at their disposal. I often wonder what their reactions would be to see the sophisticated sensors we have available to us today.

As usual, my poor old noggin is full of random thoughts bouncing around like super balls on steroids. The topic that has currently captured my attention is that of sensors.

What pops into your mind when you hear (or see) the word 'sensor'? Living in an age of wonders as we do, you may be thinking of highfalutin' devices like lidar (laser detection and ranging) or radar (radio detecting and ranging). When we boil things down, however, a sensor is any device that detects some physical phenomenon and produces a corresponding output signal.

For our purposes here, we will assume electrical output signals in terms of voltage or current being used to feed electrical or electronic systems, but this isn't cast in stone.

Victorian fax machines

Can you imagine the Victorians sending faxes to each other? This may seem far-fetched, but in 1842, a Scottish engineer and inventor called Alexander Bain came up with a cunning idea. He created an image to be transmitted by cutting it out of a thin sheet of tin. He placed this metal image on a movable insulated base and connected it to one side of a battery. The base was slowly passed under a swinging pendulum formed from a conducting wire with a weighted point on the end. Whenever this point connected with the metal image, it completed the electrical circuit, thereby converting the dark and light areas of the image – which were represented by the presence or absence of tin – into an electrical signal.

This cunning creator used this electrical signal to activate a relay attached to the end of another pendulum that was swinging back and forth over a second moving bed. The activated relay caused an attached pencil to encounter a piece of paper laying on the moving bed, thereby reproducing the original image in metal as a drawing in pencil.

Ray guns and TV controllers

Did you ever see the original *Buck Rogers* science fiction serial from 1939 starring Buster Crabbe? Filmed in glorious black-and-white, this was originally released as

a series of 20-minute movies, and later shown in the 1950s and 1960s on TV.

Buck sported a ray gun in the form of a 'U-235 Atomic Pistol'. The reason I mention all this is that the first practical photoelectric cells were invented in the 1880s. In 1955, Zenith introduced the world's first wireless television remote control called the Flash-Matic. Looking like something Buck Rogers would not be ashamed to be seen carrying, this glorified torch (flashlight) employed a beam of light to activate four photocells located at the corners of the screen, thereby allowing the user to control the volume and channel selection.

Mobile sensor platforms

Do you remember the artificial intelligence (AI) called KITT (Knight Industries Two Thousand) powering the highly advanced, very mobile, robotic automobile in the *Knight Rider* TV series of the 1980s? Today's cars are getting close to (sometimes they surpass) KITT's capabilities.

My own 2019 Subaru Crosstrek is equipped with binocular vision that can be used to detect and correct any drifting out of lane, vary the speed of the cruise control if we get too close to a car in front, and slam on the breaks if it feels we are in danger of imminent collision. All of this is made even more exciting by my wife screaming in my ear.

In fact, today's autonomous cars and robots are essentially mobile sensor and computing platforms. A very common scenario is to have multiple cameras equipped with CMOS sensor arrays that are sensitive to light in the visible part of the spectrum. These feed advanced processors running AI algorithms that can perform tasks like object detection and recognition.

These cameras can be augmented by lidar and radar sensors. The original lidars were big, bulky, and expensive, but new versions are coming online in which almost everything is implemented in semiconductor form. As opposed to a simple time-of-flight (TOF) approach which involves generating powerful pulses of light and measuring the round-trip time of any reflections, companies like SiLC Technologies are using a frequency

modulated continuous wave (FMCW) approach that can provide distance and velocity data on a pixel-by-pixel basis, allowing them to perceive and identify objects more than a kilometer away.

Meanwhile, companies like Owl Autonomous Imaging are creating long-wave infrared (LWIR) thermal focal plane arrays (thermal imagers). The signals from these images can be employed by AI to perform object detection, classification and ranging. As the folks at Owl told me, 'Within five years, all new cars will be able to see at night!'

That's deep!

Have you ever thought about our amazing ability to perceive the world around us in three dimensions? Powered by our optical sensors (eyes) and associated computers (brains), we call this ability 'depth perception.' There are many aspects to this, but we start with the fact that each of our eyes sees a slightly different image due to their separation in our heads. The resulting disparities are processed in the visual cortex of our brains to yield depth information.

Even with one eye closed, we can still do things like track and catch a ball heading our way. In this case, our brains make use of visual cues, including knowing how big we expect objects to be, and our understanding that if an object appears to be growing bigger, then this may be a good time to duck.

In the case of machine vision, one of the components of depth perception is the ability to create a 3D depth map (point cloud) of the scene. We can do this using two CMOS sensors to provide binocular vision, but that increases the cost. We can employ a single CMOS sensor in conjunction with an AI, using its understanding of the scene to determine where and how big things are in 3D space, but this requires a lot of computation.

The folks at a company called AIRY3D have come up with a way to use a single CMOS sensor to generate both a regular 2D image and a 3D point cloud on a pixel-by-pixel basis with very little computation. I don't know about you, but I certainly didn't see this one coming!

Win a Microchip PolarFire SoC Discovery Kit

Practical Electronics is offering its readers the chance to win a Microchip PolarFire SoC Discovery Kit (MPFS-DISCO-KIT) – and even if you don't win, receive a 15%-off voucher, plus free shipping for one of these products.

The PolarFire SoC Discovery Kit features a quad-core, RISC-V application-class processor that supports Linux and real-time applications, a rich set of peripherals and 95K of low-power, high-performance FPGA logic elements. This fully featured, yet low-cost kit allows the rapid testing of application concepts, development of firmware applications, programming and debugging of user code.

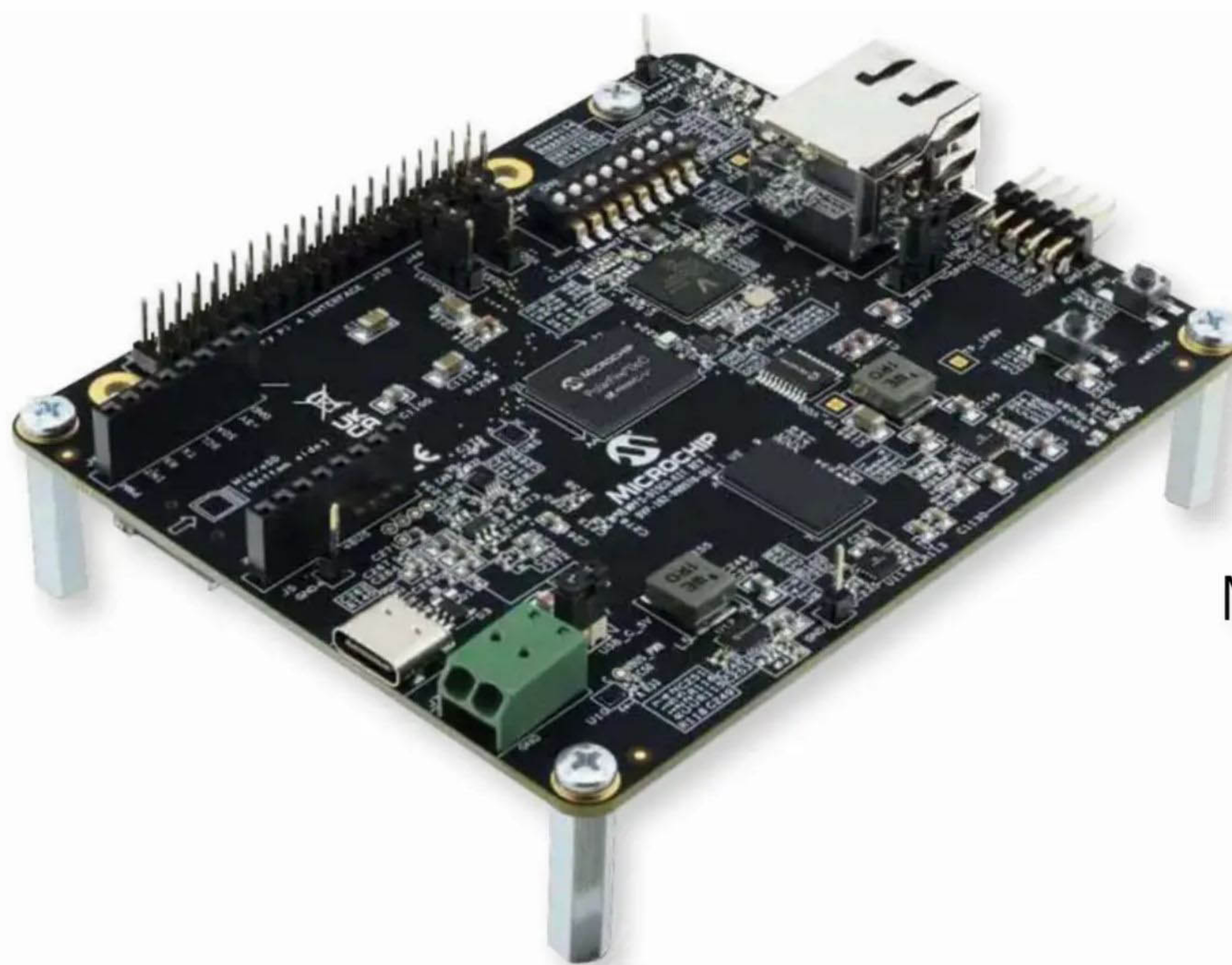
The Discovery Kit is built around the PolarFire MPFS095T SoC FPGA that features an embedded microprocessor subsystem consisting of a quad-core, 64-bit

CPU cluster based on the RISC-V Instruction Set Architecture (ISA). A large L2 memory subsystem can be configured for performance or deterministic operation and supports an asymmetric multi-processing (AMP) mode.

**Worth
\$132.00
(approx £104.50)
each**

The board includes support for Microchip's Mi-V ecosystem, a MikroBUS expansion header for Click Boards and a 40-pin Raspberry Pi connector, as well as a MIPI video connector. The expansion boards can be controlled using protocols like I²C and SPI. One GB of DDR4 memory is available, as well as a microSD card slot for booting Linux. Communication interfaces include one Gigabit Ethernet connector and three UART connections via the USB type C connector. An on-board FlashPro5 programmer is available to program and debug the PolarFire FPGA through a USB-to-JTAG channel.

Free-to-enter competition



Microchip

Win a Microchip PolarFire SoC Discovery Kit

Microchip
MPFS-DISCO-KIT

How to enter

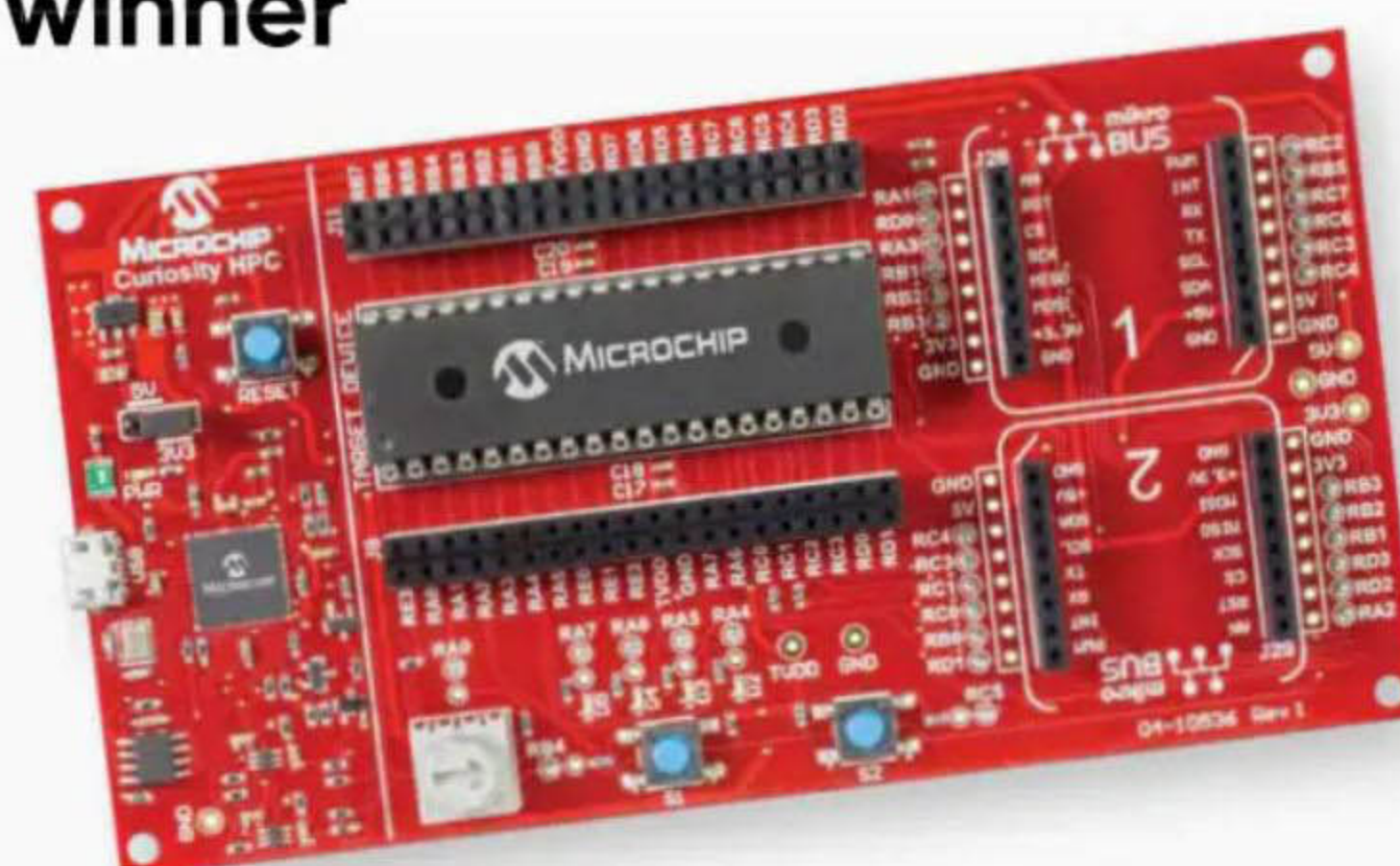
For your chance to win a Microchip PolarFire SoC Discovery Kit or receive a 15%-off voucher, including free shipping, enter your details in the online entry form at:
<https://page.microchip.com/PE-SoC.html>

Closing date

The closing date for this offer is 31 July 2024

June 2024 winner

David Izzard



David won a Microchip Curiosity HPC Development Board

Net Work

Alan Winstanley

This month's column explains how the use of passkeys improves online security and eliminates password logins. The author also transfers data to a new phone and shares some hints and tips.

Every web user will recognise the annoying requirement to log into any website that retains a user's personal details. Even trivial online transactions usually involve inputting our name, address, mobile number and email address. Some even impertinently ask for our date of birth, or they might offer to remember debit and credit card numbers as well, an offer I never accept!

Much has been written about the need to use a mix of special characters that makes passwords harder to guess. The old idea of substituting alpha characters with numbers (called 'leetspeak') such as p455w0rd has fallen out of fashion, and probably the best practical advice is to use a 'passphrase' of several words and maybe a special character at the end, such as: bread-coffee-7281#. A longer string of random words makes passphrases even harder to guess, for example: tulip.sugar.lawn.yoyo.

To help users to memorise their passwords, I've noticed how special notebooks are sold, helpfully labelled 'Password Book' or similar, which would undoubtedly be goldmines in the wrong hands. Recording logins and PINs openly in clear print this way would also breach the terms and conditions of banks and credit card providers, rendering the account holder entirely liable for any losses caused by fraud. If you can't remember logins, and you're maybe

liable to forget where you last left your password book, they're not a good idea.

A limited choice of online password managers is available that automates the task of logging in or filling in online forms. In the author's case, having accumulated over 400 different logins over time, online security has been managed for 15 years with Roboform password management software. Roboform has been established 25 years and is something of an unsung hero in my opinion, as without it, life at a busy desk would soon grind to a halt; it's one of the very few online services that I'm happy to invest time and money in, and it also works across mobile devices. It's a powerful programme with many features and a few quirks, and you soon wonder how you managed without it.

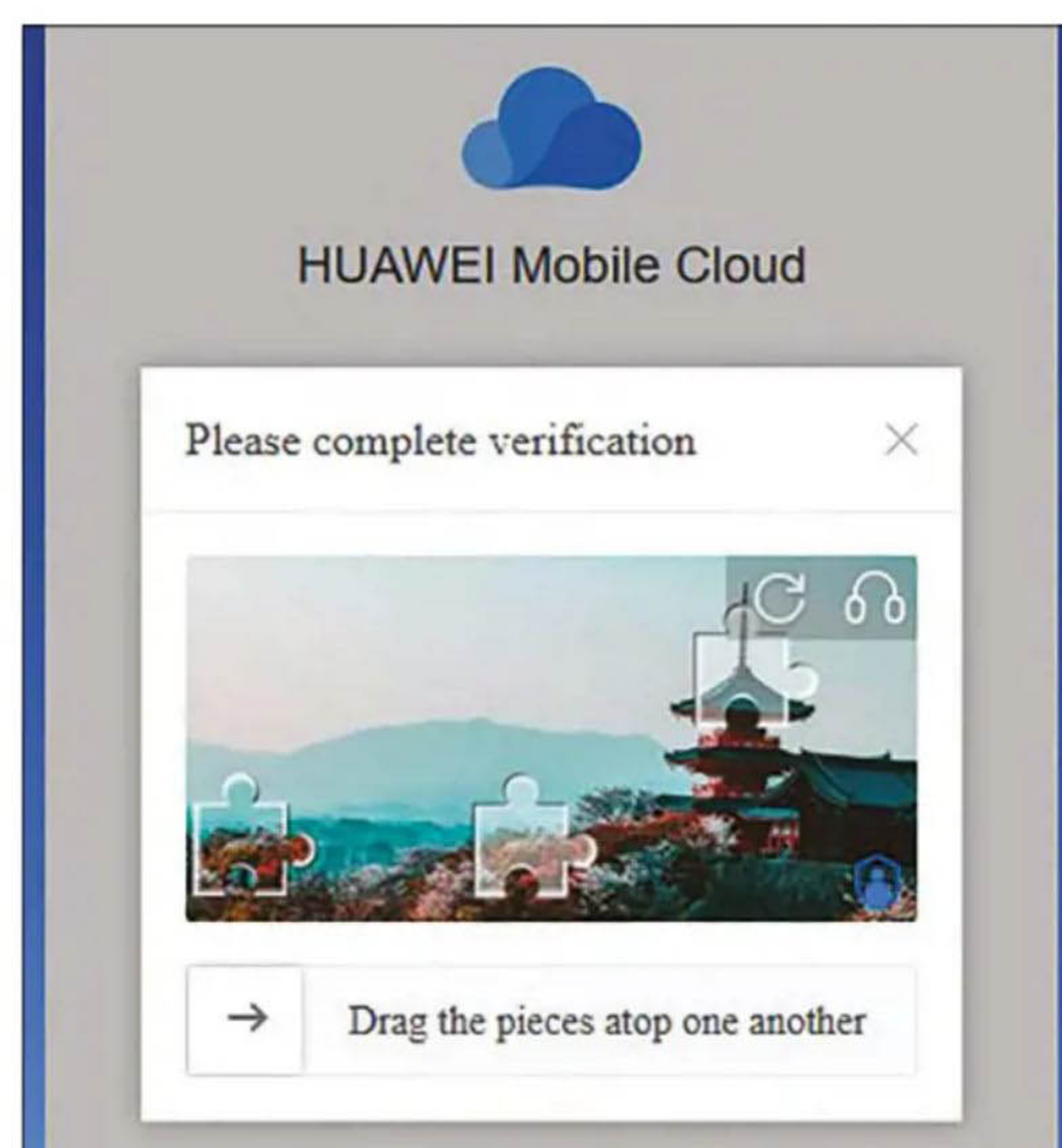
Another popular password manager is LastPass, a product that has suffered at least two security breaches in the recent past. The Tech Republic website doesn't recommend LastPass for that reason alone, and tech website The Verge reckons LastPass may 'have a lot of work to do if it wants people to trust it again'. LastPass is striving hard to restore credibility and has now implemented a minimum standard of 12-character passwords. Others to try include highly-rated NordPass (<https://nordpass.com>) and 1Password (<https://1password.com>). Obviously, you must also safeguard access to your PC and any password manager software as well, so explore those security options carefully.

Crumbling cookies

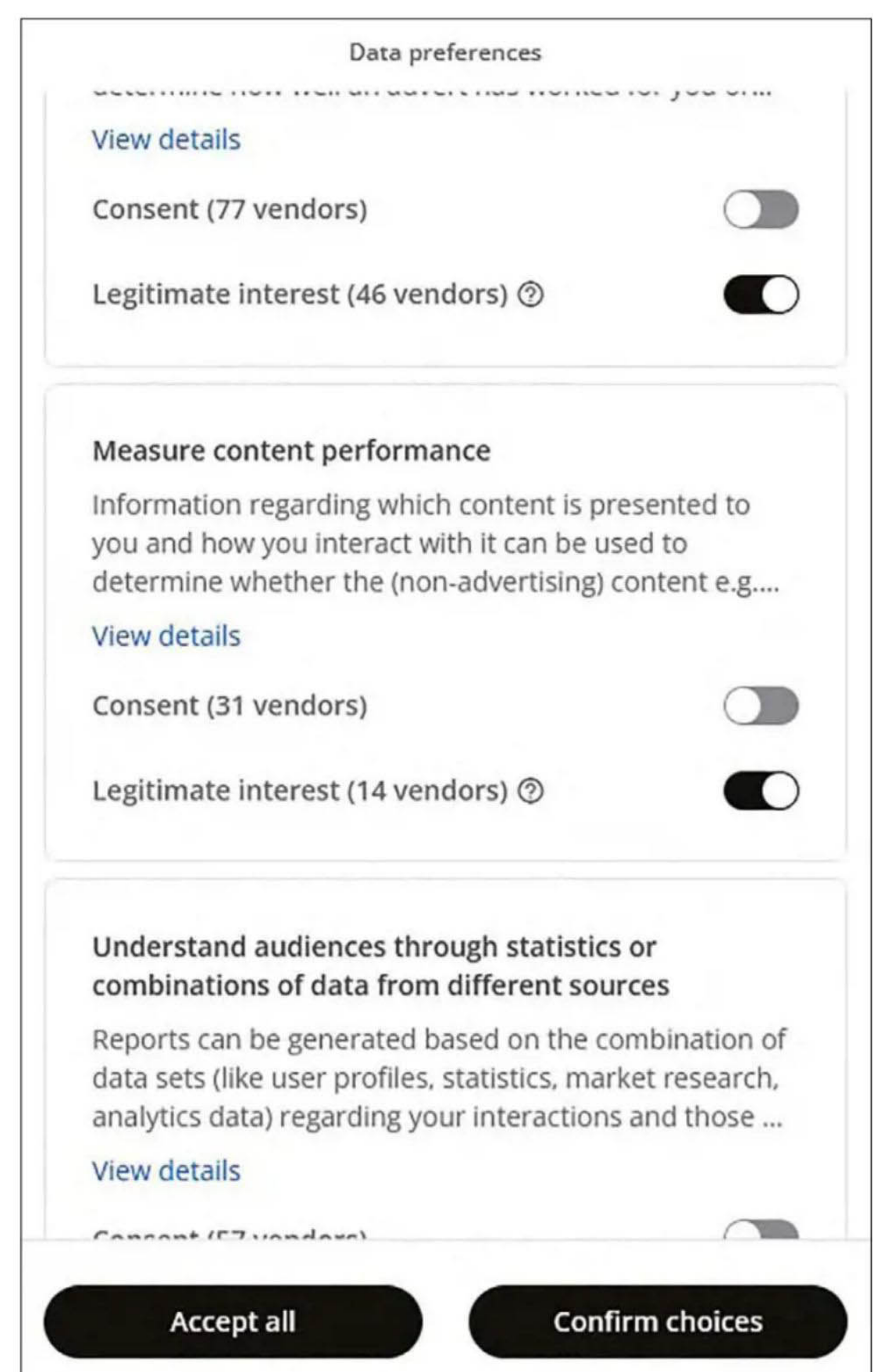
During the log-in process, users are often interrupted by a 'Captcha' screen, which aims to prevent automated scanning or hacking of a website by 'bots'. A Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) 'challenge-response' causes a user to physically respond to a question, supposedly proving that a real human is logging in.

Common Captcha screens ask 'how many traffic lights' or 'what is four times two', or they may ask users to drag a jigsaw puzzle piece into position, but advanced hacking can nevertheless still defeat some of them. Admittedly, even I struggle to make sense of some Captcha screens, but, sadly, the ones who struggle most with this hurdle are those with vision problems who rely on screen reader software to surf the web. Web page screens that say 'I am not a robot' might seem deceptively simple or pointless, but they're designed to be screen-reader friendly, and Google's reCAPTCHA v3 doesn't need a challenge at all.

Another log-in annoyance is the cookie opt-out screen, necessitated by the European GDPR and ePrivacy Directive. Cookie screens have spread like a rash across the web, and seem intended to punish website visitors seeking to safeguard their own privacy. Standardised cookie modules are often



This Captcha screen uses a sliding jigsaw puzzle piece to confirm you're really human.



Some cookie opt-out screens make it onerous for users wanting to opt out of receiving unwanted cookies.



'Take Five' is a UK Government campaign with lots of resources to highlight the risks of scams, fraud and identity theft.

used on websites. The smartest ones have a simple 'Reject All' or 'Essential only' button allowing users to quickly opt-out before proceeding, while others annoyingly require visitors to opt out of a dozen 'legitimate interest' cookies, one by one. As often as not, this latter type gets a thumbs down from the author and I will simply go elsewhere instead.

As an aside, web surfers will often see click-through URLs containing strings of characters that register web server statistics (Google Analytics) – an example might be: https://anywebsiteurl.com/?utm_source=google&utm_medium=emaillist&utm_campaign=128465419&utm_content=468754&utm_term=cookietest&ad_source=2&gclid=Cj... etc.

The term 'UTM' means 'urchin tracking module' which is Google's website analytics at work. The string might include search terms, your username or email address in plain text or other data that identify trends and visitor behaviour in the stats. Facebook does the same, and the string `...url/?fbclid=blah...` often appears in website tracking code, containing the Facebook Click Identifier. These days, I usually cut and paste these URLs into Notepad and delete the question mark and everything after it, in order to defeat Google Analytics. Only then do I paste the shortened URL into my browser.

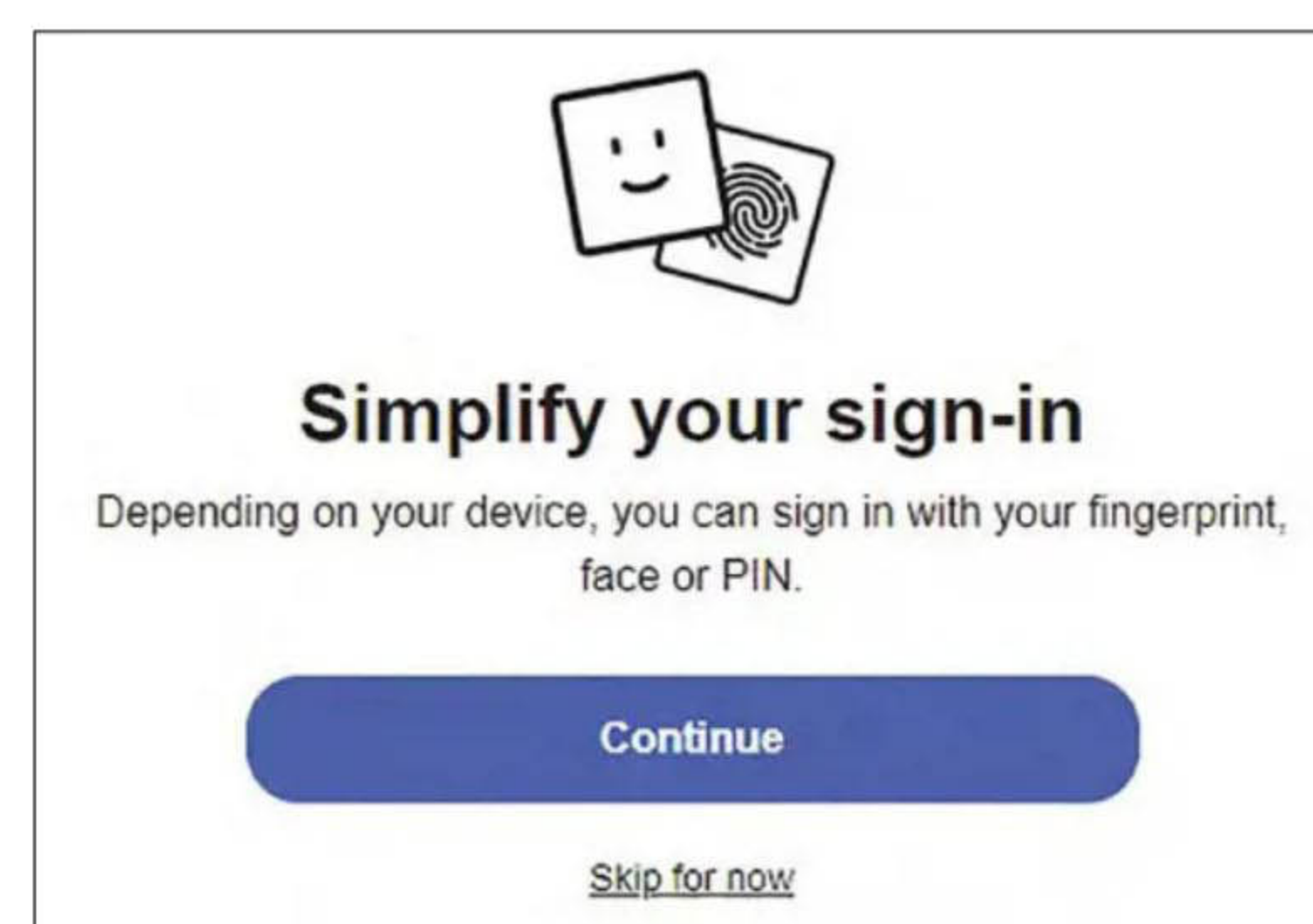
Apart from being hard to remember, another major problem with passwords is that, despite using multiple words and special characters to defeat hackers, they're inherently insecure in the first place, simply because users can be tricked into revealing them through the use of fraud or phishing scams, for instance. Or maybe that Password Book could be stolen or lost.

A phishing email or a highly targeted spearphishing mail may trick a victim into visiting a fraudulent website, where the password can be captured by crooks, or ransomware could be downloaded onto a visitor's computer. Some phishing emails can be highly convincing, especially if they are timely ones (eg, one supposedly from 'DHL' arrives when I really am waiting for a DHL delivery), and even the author has had to hit the brakes once or twice before clicking a likely-looking link. It's so easy to fall for this type of fraud when busily working online; as the UK's National Cyber Security Centre (NCSC) says, 'Asking users to examine, in depth, every email they receive will not leave enough hours in the day for work tasks. It's an unrealistic and counter-productive goal because responding to emails and clicking links is an integral part of work.' The Government advice to guard against these threats is to 'Take Five', see: www.takefive-stopfraud.org.uk

Take Five contains many educational resources, web banners and information that can help to spread the word: if you're involved with a local Facebook group or have vulnerable friends or relatives, the Take Five campaign is a timely reminder of these potential risks and it's worth directing them to it.

No pesky passwords – get a Passkey

In extreme cases, passwords or user data may be stolen from a website's database following, say, a ransomware attack, and within a few seconds they can fall into the hands of fraudsters anywhere in the world. Some websites are beginning



Major websites such as eBay offer to create passkeys for users – they're far more secure and easier to use than passwords, and are worth setting up on your device.

to offer alternative logins using much more secure *passkeys*. These are fully encrypted and use a unique 'private' key on the user's device together with a corresponding 'public' key stored by an online service. Old hands will recall PGP (Pretty Good Privacy) which was one of the few ways of encrypting or signing messages or transactions using PGP keypairs. Put simply, it's rather like a user giving out open padlocks (the public key) to all and sundry, asking them to lock up messages with them before delivering them to you. It doesn't matter who has an open padlock, because you're the only one with the (private) key who can open them again. The private key is then safeguarded on your device by a PIN, a password or biometrics like fingerprints or facial recognition. Accessing the private key also proves that you are physically in control of the device on which it is stored.

Websites including PayPal, eBay and Amazon now offer to set up passkeys as logins and they are an excellent idea for safeguarding security. Passkeys also make logging in from a device much simpler, as there's no password to remember. Amazon has a useful primer on how passkeys work at: <https://bit.ly/pe-aug24-amz>

A token gesture

Other methods of enhancing security include the use of physical devices



HAMMOND®

1554/1554F Polycarbonate IP68

Learn more:
www.hammondmfg.com/1554
uksales@hammondmfg.com • 01256 812812



new
sizes!





These Yubico security keys provide a physical ‘touch token’ that protects your logins securely against fraud or ID theft. Both USB-A and USB-C types are available.

or ‘tokens’ such as the USB security keys produced by Sweden’s Yubico. These are used by all Google’s staff and contractors for secure computer and server logins, Yubico says. I covered Yubico’s secure keys in the June 2020 issue of **Net Work** and more details of these hardware devices are online at: www.yubico.com

The author’s Facebook account is secured with a Yubico USB security key which, as an ID confirmation check, needs a simple touch-tap to prove that I’m physically present when logging in. Other higher-security types have a proper biometric fingerprint reader built in, but they become pricey – around €95, exc. tax. Yubico keys are available in USB-A and USB-C styles, and a tiny ‘Nano’ version is designed to reside in the port.

The buzzword here is ‘FIDO authentication’ which is a powerful encryption protocol at the heart of Yubico secure keys. You can learn more about FIDO at: <https://bit.ly/pe-aug24-fido> and a catalogue of products and services that are ‘Yubico-aware’ is at: <https://bit.ly/pe-aug24-yub>

Over time, I expect to see more acceptance of hardware tokens like these as online security becomes ever more challenging.

Time to swap phones

Like many people, the writer has come to rely on a smartphone for helping with many everyday tasks or keeping track of communications. In some applications they are virtually indispensable: I wish it wasn’t always the case, but services including routine banking, shopping or even buying a parking ticket now use apps and, at the very minimum, consumers are often expected to have a mobile phone to receive security codes

– One-Time Passwords (OTPs) – sent by financial institutions or service providers. Just to reiterate a vital point: OTPs that you receive should never be given out to anyone else, as they may well be fraudsters looking to steal from you. If you unexpectedly receive an OTP, it may be a sign of fraudulent activity taking place, so be on your guard and investigate if necessary. An authenticator app on a smartphone can also be used to generate a code number as part of the 2FA process – both Google and Microsoft offer them.

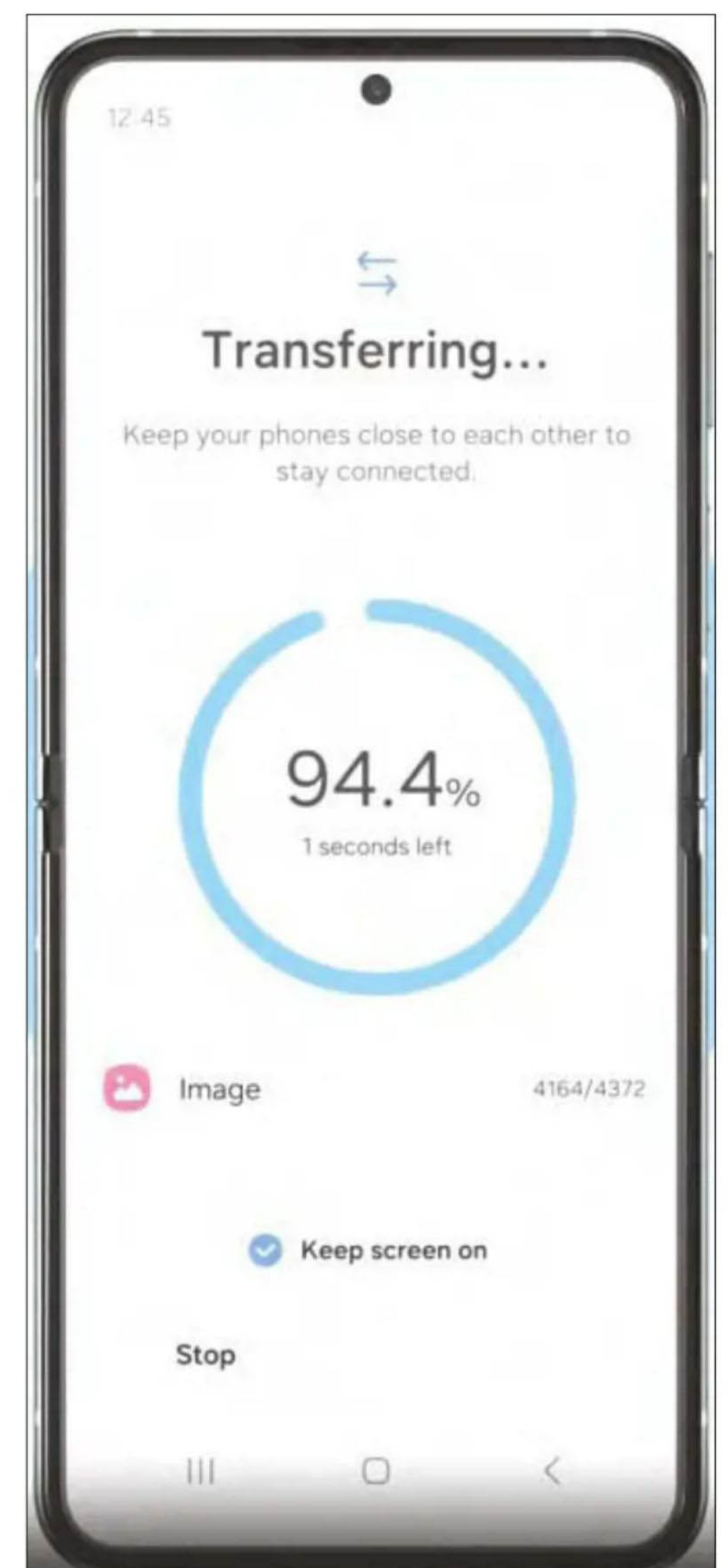
In the past few weeks, I finally upgraded my Huawei P20 smartphone as it was showing its age and was starting to buckle under the workload. It must be said that the Huawei has proved faultless, and it was generally a pleasure to use, but US sanctions against the brand effectively killed off the UK market, and so the next task was to transfer its contents to a new Samsung Galaxy 5G. I’ll summarise my experience of upgrading a phone, with hints and tips that I hope will help readers to prepare for the same eventuality.

Although I rather dreaded the task, in reality, the job of moving everything over to a new phone went exceptionally well. For good measure, a cloud backup of my thousands of photos and video files is offered by Huawei and is still available for as long as I pay the negligible annual running cost. Huawei has kept its side of the bargain, and the cloud backup works efficiently, so I downloaded the cloud backup onto my PC as an extra measure.

Switch it up

On powering up the new phone, a setup routine asks whether you want to transfer data to it from another device, and Samsung’s Switch app is duly installed. The same app is also needed on the

old phone, after which the process of copying contacts, mail, apps and media – including all those photos and videos – started automatically. This can be handled over Wi-Fi or via USB. Note that iOS as well as Android are supported. Happily,



The Samsung Switch app worked flawlessly when transferring data and apps from an old mobile phone to a new Samsung smartphone.



You can safeguard a smartphone camera lens with a low-cost tempered glass protector.

the process was very simple, and no particular problems were experienced, but be ready to input those pesky usernames and logins when launching them on the new phone for the first time. WhatsApp, which is encrypted end-to-end, will only operate one account on one device at a time, so the ‘old’ account was automatically disabled on the old phone, but all WhatsApp media were migrated effortlessly to the new device.

What I thought would be the trickiest transfer of all – the HSBC online banking apps – was totally seamless; HSBC allows you to use up to three devices and a QR code helped complete the operation very smoothly, with no issues experienced at all. As another benefit, the much better camera on the Samsung ‘snaps’ QR codes with just a cursory glance (see later). Everything went commendably smoothly, and I reflected that, finally, modern technology had delivered on its promises. Samsung, Google and Microsoft will jostle to offer cloud backup plans as well, and it’s worth spending time getting to know at least one of them.

Thumbs up

Next, biometrics can be set up as your fingerprint ‘dabs’ must be scanned by the new device for the first time. This was a slow process, and it’s wise to scan multiple fingertips in case one suffers cuts or abrasions. Samsung allows up to four prints to be scanned, but this idea won’t work if fingertips are wet through rain or moisture, so it’s definitely worth remembering PIN number alternatives and do practise using them to remain

familiar with the code. Facial recognition scanning is another biometric option, but I didn’t bother setting it up.

It was then a matter of getting to know the new phone. Tempered glass screen protectors costing just a few pounds are sold on eBay which safeguard against damage. A seller called ‘Pixfab’ supplied mine at low cost and it fitted perfectly thanks to the kit containing a screen wipe and drying tissue. It’s a one-shot operation so align everything carefully before releasing it onto the screen. A couple of ‘bubbles’ disappeared by themselves after a few hours and the screen protector is totally invisible. (An option in Settings can increase touch-screen sensitivity if screen protectors are used.) A tempered glass protector for the triple camera lens array, also supplied by Pixfab, fitted perfectly as well. The biggest headache was probably finding a suitable case, as the web is awash with Chinese-made products, but one branded ‘Qltypri’ proved fine, without spending silly money. What is comically marketed as ‘PU leather’ is – of course – actually polyurethane!

The experience taught me that it’s feasible to set up a cloud backup, especially useful in case your phone is stolen or lost, with data uploaded over Wi-Fi rather than using mobile data. It’s critical to keep on top of account logins too, as these are needed the first time apps are opened on the new device. For anyone interested, Samsung fully explain the details of the Switch app, giving readers a good insight into compatibility and how it works – see: <https://bit.ly/pe-aug24-sam>

Back on the topic of QR codes; as far back as the February 2012 issue I described how these new pixelated peculiarities would transform the way we captured data, by using our camera phones as scanners. As I explained above, I found my new smartphone scans even large, complex QR codes instantly, but it’s worth remembering that *fake* QR codes sometimes appear in public spaces. Counterfeit QR codes might sometimes be stuck over genuine labels: one car park was covered with them, with scammers trying to rake in cash from motorists. Railway station car parks and posters are another prime target. The UK’s NCSC offers advice which is worth recalling before you go and ‘snap’ a likely-looking QR code: <https://bit.ly/pe-aug24-ncsc>

Fun with FLIR

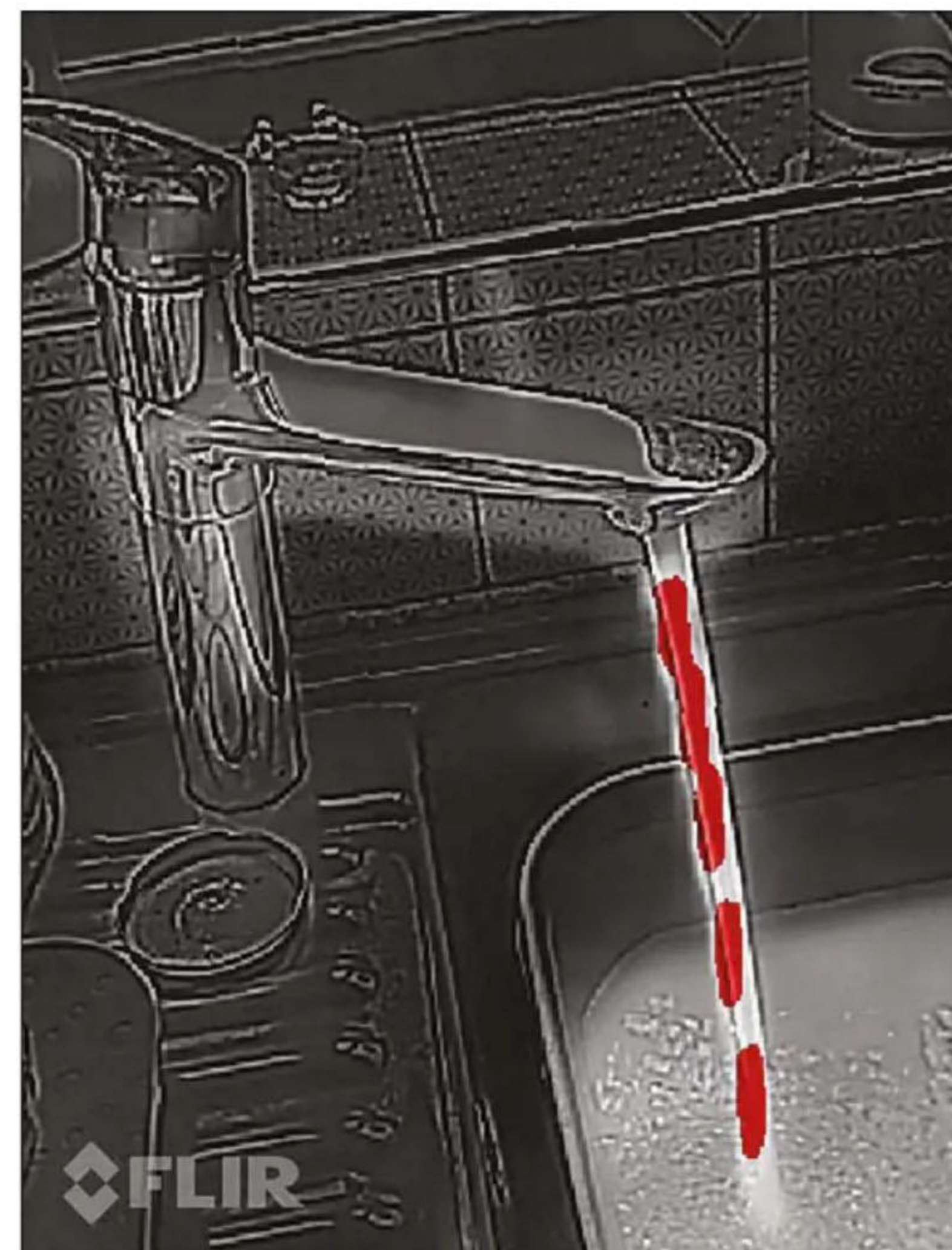
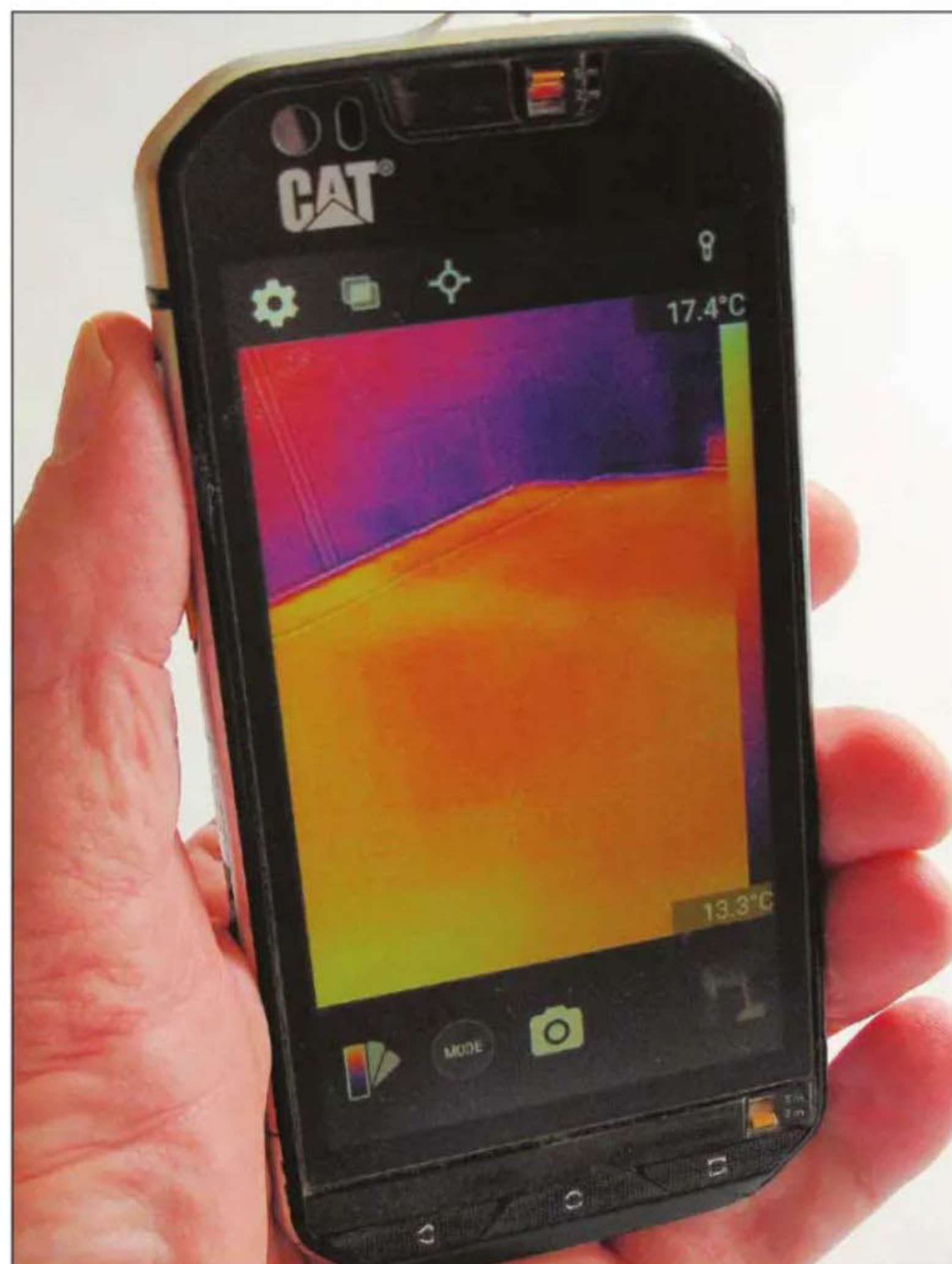
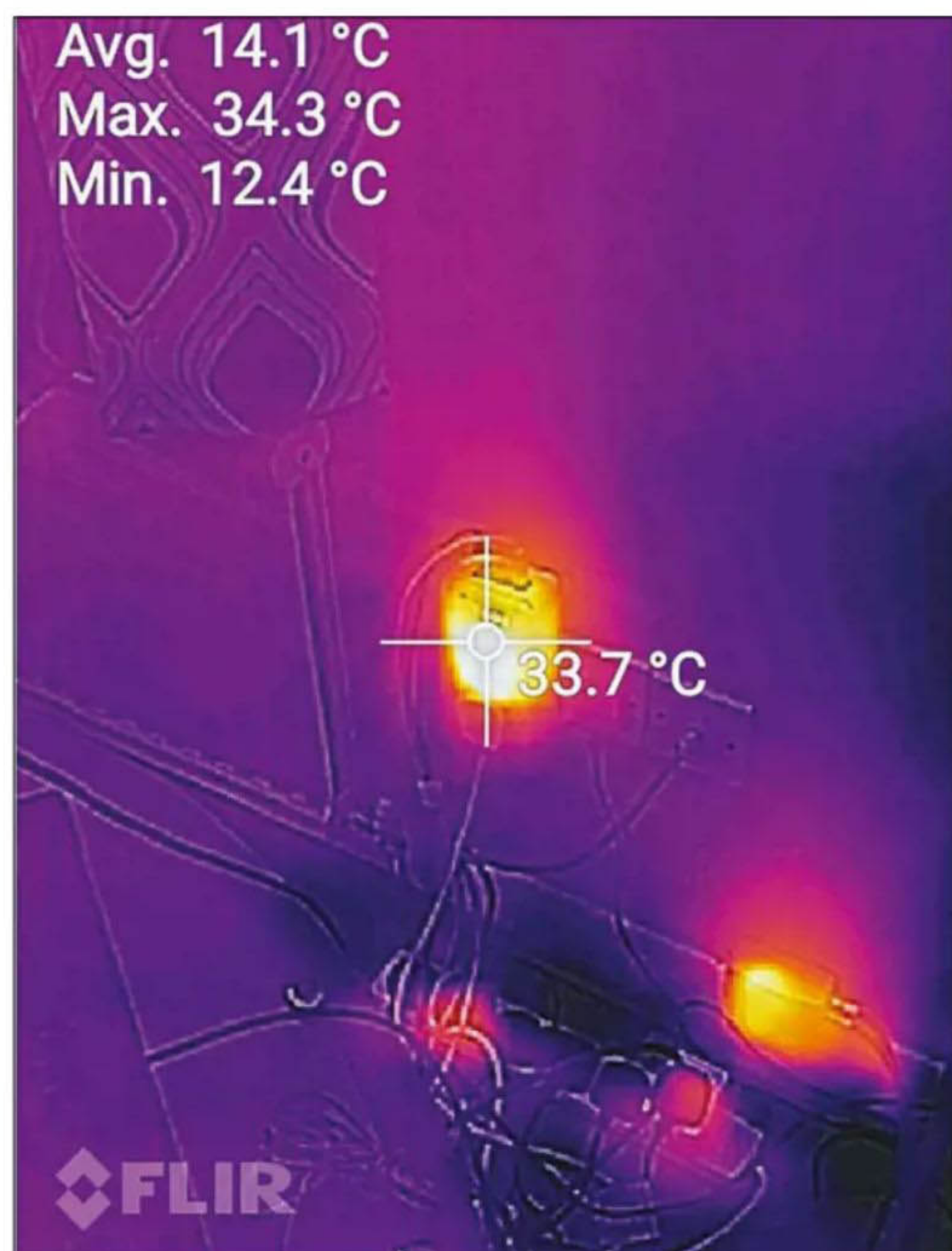
The July 2024 issue of *PE* included a design for a Pico-based thermal camera which, compared with the cost of commercial units, is an inexpensive and worthwhile introduction to capturing infra-red heat maps and exploring the emissivity of materials. My computer colleague recently dropped into the office with another toy to play with: a ‘CAT’ branded ruggedized smartphone with a difference, as it has a built-in FLIR (Forward-Looking Infrared) thermal camera. The CAT S60 battery was on its last legs, but I managed to power it up with a USB powerbank. The



- Project boxes designed and manufactured in the UK.
- Many of our enclosures used on former Maplin projects.
- Unique designs and sizes, including square, long and deep variations of our screwed lid enclosures.
- Sub-miniature sizes down to 23mm x 16mm, ideal for IoT devices.

www.terrington-components.co.uk | sales@terrington-components.co.uk | Tel: 01553 636999





This toughened ‘CAT’ smartphone includes a FLIR thermal camera. It can also highlight hot (or cold) spots and heat emitted by ‘vampire’ gadgets.

FLIR camera can capture heat maps in colour, as well as imaging in monochrome and highlighting hot spots in red, or cold spots in blue. Spot-readings of temperatures are displayed. Checking a hot water supply or looking for ‘vampire’ electrical gadgets is fascinating. FLIR phones crop up on eBay from time to time, and as our project designer agreed last month, thermal cameras could help diagnose trouble spots in all sorts of situations. As for replacing the CAT’s battery, well, that’s a project for another day!

Aiming for the stars

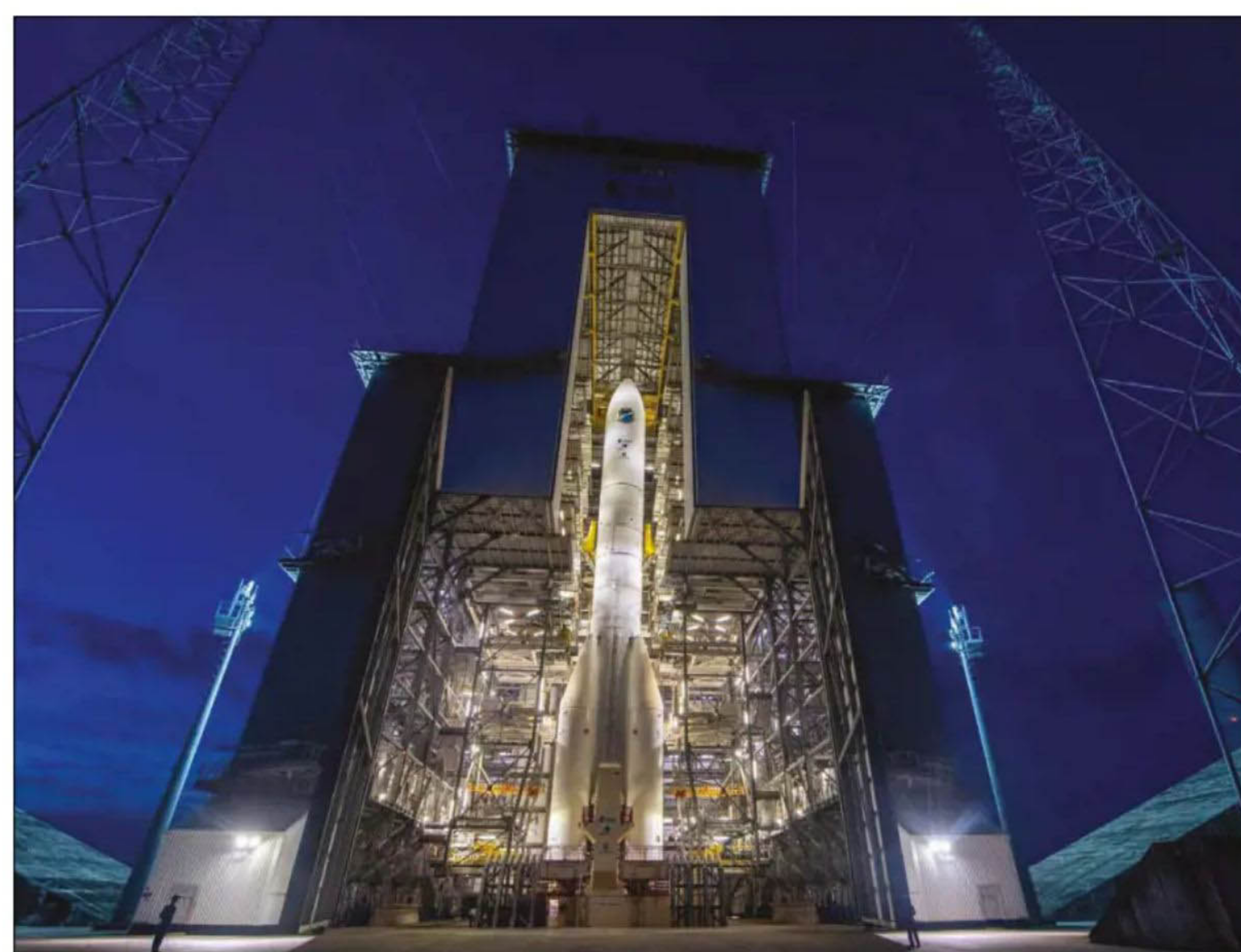
Next, a roundup of current space missions and projects from around the world. China’s Chang’e 6 probe, launched in May, successfully touched down on the far side of the moon, and it’s hoped soil samples will eventually be returned to Earth. After a false start or two, Boeing’s Starliner launched its first crewed flight for NASA on 5 June, docking successfully with the International Space Station on a mission initially intended to last about ten days. A number of small helium leaks in the capsule are being investigated, and the next mission will carry three or more astronauts sometime in 2025.

Sierra Space and NASA are now testing the ‘Dream Chaser’ spaceplane ready for launching later this year. The vehicle is the first ever uncrewed winged ‘space-shuttle’ style vehicle to be manufactured commercially (see *Net Work*, August 2023). The maiden flight of the 30-foot long (9m) vehicle, named Tenacity, will deliver cargo to the ISS using its ‘Shooting Star’ module. Sierra Space also has ambitions to build the first commercial ‘inflatable’ space station in the future.

The inaugural flight of the European Space Agency’s long overdue Ariane 6 heavy lifter is scheduled for the 9 July, launching from French Guiana on the South American coast. A key feature of Ariane is that its upper stage main engine can stop and start up to four times, allowing it to deliver payloads at four locations along its trajectory, bus-stop fashion, rather than having to release an entire payload all in one go.

The ESA has striven to replace Soyuz launches with Ariane ever since co-operation with Russia ceased following the Ukraine invasion. Space fans can, however, marvel at the Soyuz ‘User’s Manual’ still available on the ESA website at: <https://tinyurl.com/mv7zpukd>

It contains a fantastic wealth of detail plus a history of USSR Soyuz rockets, starting with Sputnik’s launch 67 years ago, an



The first test flight of the European Space Agency’s Ariane 6 heavy lifter is scheduled for early July. (Image: ESA)

accomplishment that heralded the start of the space race, and in turn saw the creation of a new, resilient, self-healing packet-switching data network – what became known as the Internet.

And finally

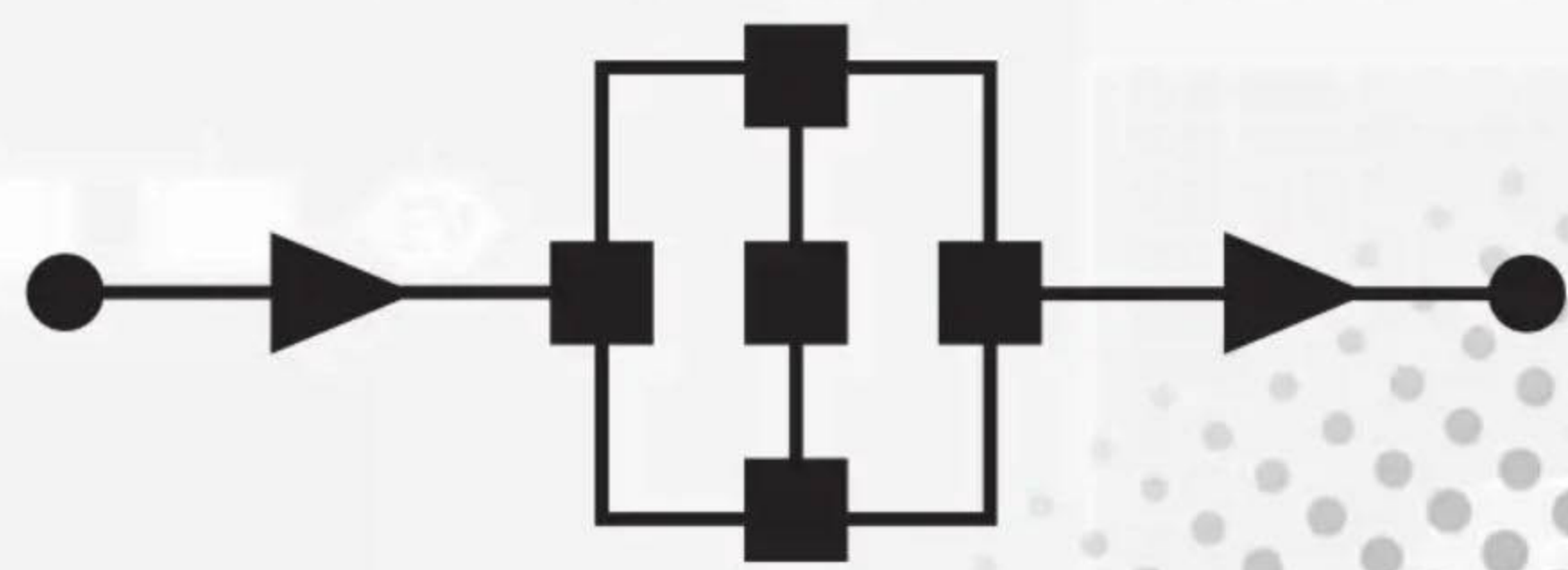
The arrival of the 1990s ‘Internet’ for consumers is where I came in, with the first column appearing in August 1996 when the world-wide web barely existed. This month’s *Net Work* is sadly the last one to appear under the auspices of the current Editor and Publisher, Matt Pulzer. Matt has edited *PE* successfully for many years and also became the Publisher in 2018, and has worked tirelessly to bring *PE* readers the widest choice of projects, tutorials and interesting topical features. Navigating us safely through the stormy waters of pandemics and lockdowns, Matt’s dedication and expertise has given each issue a touch of finesse and class. With Matt’s encouragement *Net Work* grew into a feature bringing readers news and trends covering the Internet, technology, space, energy and more. Matt explains elsewhere what’s in store for your favourite hobby electronics magazine, so here’s my personal ‘thank you’ to Matt for supporting *Net Work* throughout all these years, and more importantly, for bringing readers their copy of *Practical Electronics* every month. Good luck, Matt!

See you next month for the latest from *Net Work*!

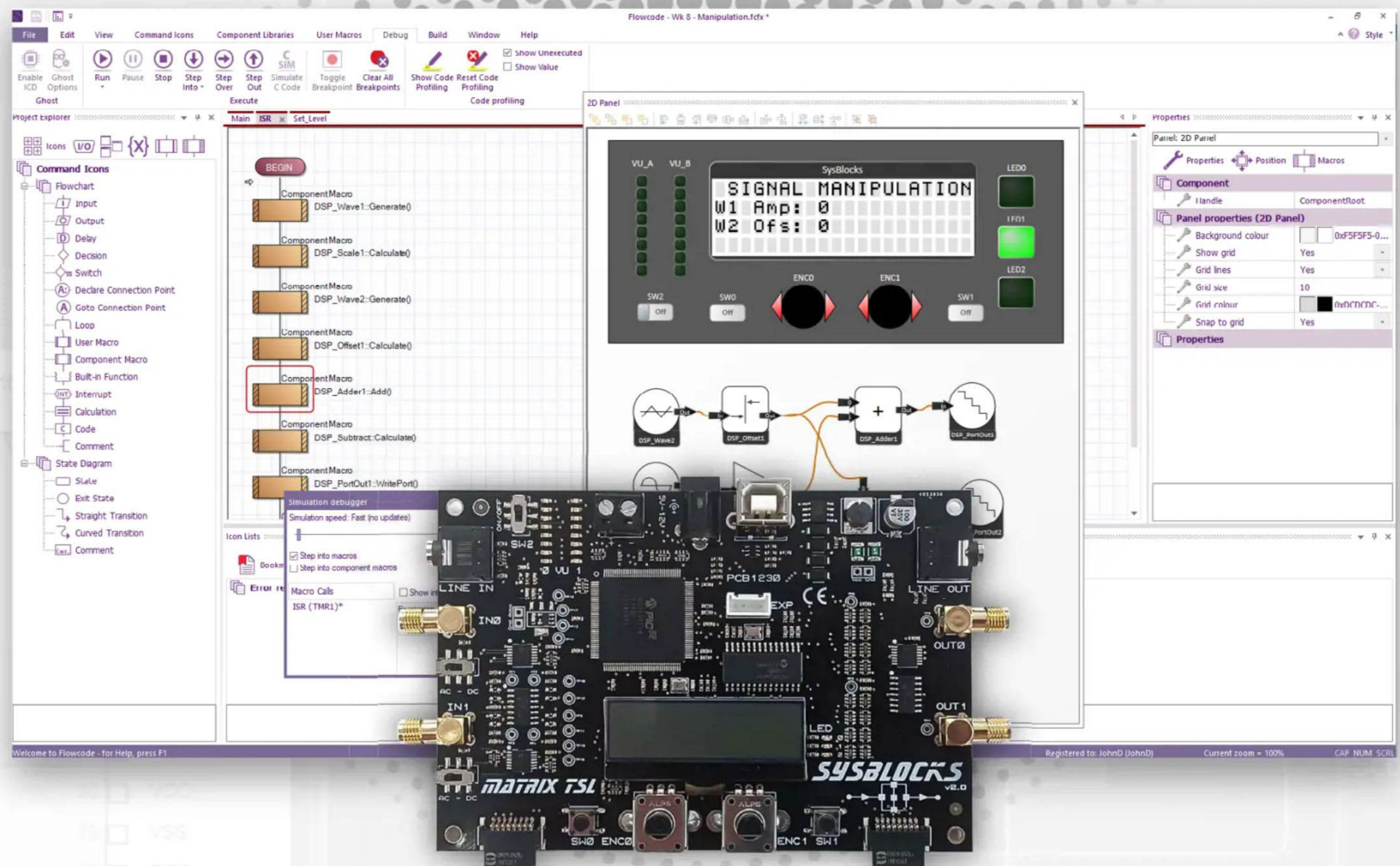
The author can be reached at:
alan@epemag.net

Teach-In Check Point answers – see page 66-67.

1. c 2. b 3. b 4. c 5. c 6. b 7. b 8. a 9. b 10. c
11. a 12. c 13. a 14. a 15. c 16. a 17. b 18. b 19. a 20. a



Sysblocks



CREATE HIGHLY FUNCTIONAL ELECTRONIC SYSTEMS USING PIC32 TECHNOLOGY TO MANIPULATE ANALOGUE SIGNALS USING FLOWCODE 10.1 AND SYSBLOCKS. FANTASTIC FOR DIGITAL AUDIO APPLICATIONS.

Flowcode Pro, PIC32 pack with Sysblocks board, cables and accessories bundle £610 + VAT

Flowcode with PIC32 pack with Sysblocks board, cables and accessories bundle £330 + VAT

10% off for PE readers using code: **PESYS10**

FLOWCODE 10.1 WITH MORE THAN 20 MICROCONTROLLER TARGETS INCLUDING ARDUINO UNO, PIC18877, RPI2B, AND ESP32 IS NOW FREE FOR HOBBYISTS.



+44 (0) 1422 252 380



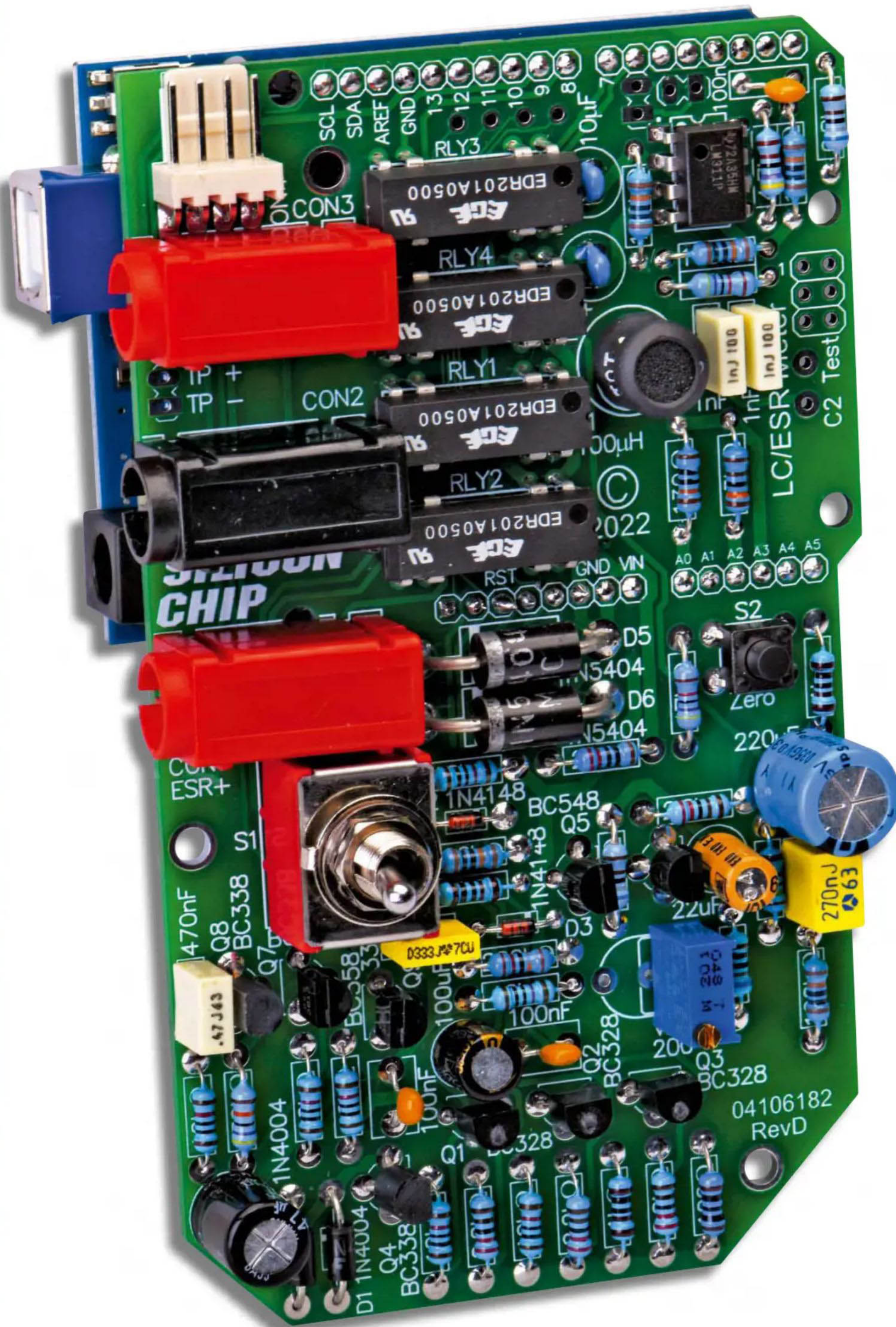
sales@matrixtsl.com

www.flowcode.co.uk

Steve Matthysen's Arduino-Based

LC and ESR METER

This enhancement to our *Wide-Range Digital LC Meter* from June 2019 allows it also to measure capacitor ESR. That is extremely useful for diagnosing faulty equipment because increasing ESR over time is one of the most common ways electrolytic capacitors fail.



Tim Blythman presented an LC Meter with excellent performance, range and accuracy in the June 2019 issue. The meter is based on a custom Arduino shield and is easy to build. Its accuracy is optimised by auto-calibration features and compensation for the inherent capacitance of the leads and even the Arduino pins.

While it's undoubtedly useful for checking suspect components, for electrolytic capacitors, it is important to know whether it has a low impedance to alternating currents. That requires it to have a low equivalent series resistance (ESR).

Why is ESR so important?

Electrolytic capacitors are used where high charge storage is required. In many applications, current must flow efficiently into and out of the capacitor to charge or discharge it. ESR acts like a resistor in series with the capacitor, losing energy each time current flows in or out.

That ESR also prevents the capacitor from doing its job properly, which is usually stabilising voltage. Say the capacitor is being charged at 1A and then starts discharging at 1A. If it has an ESR of 1Ω, the voltage seen by the rest of the circuit will suddenly shift by 2V $([1A + 1A] \times 1\Omega)$. For example, that would add to the ripple on a power supply storage capacitor.

High ESR values also lead to heating within the electrolytic capacitor, possibly changing the capacitance and reducing the integrity of its electrolyte.

One of the most common indications of failed or failing electrolytic capacitors is a sudden or gradual increase in their ESR values. Increased ESR values can introduce a wide range of mysterious circuit failures that are sometimes difficult to pin down. For a switch-mode power supply, these include decreased voltage regulation, filter failures, elevated noise levels, signal losses, or failure to start.

Therefore, it makes sense when testing electrolytic capacitors to confirm that their capacitance and ESR values are in the appropriate ranges.

Revised design

In this design we feed signals into an Arduino Uno driving an LCD. The benefit of doing this is that the ESR front end can be built on a relatively small circuit board and integrated with the LC meter presented in the June 2019 issue. That makes it a great general-purpose instrument that can

not only check the ESR of capacitors but also their values (up to a certain limit), plus it can be used to measure inductors and more.

Alternatively, you could simply attach the front end to an Arduino Uno (or clone) with a 4-line I²C alphanumeric LCD to produce a standalone ESR meter. The code for both the LC-integrated and standalone versions is available for download from the August 2024 page of the *PE* website: <https://bit.ly/pe-downloads>

If you have already built the LC meter and want to attach the ESR module, you could do that, although starting from scratch is quite possibly the easier option.

Measuring ESR

Fig.1 shows a simplified diagram representing the theory of operation. S1 and S2 are electronic switches controlled by the Arduino. When no measurement is underway, both S1 and S2 are in the discharge position to ensure the capacitor being tested and the C-Ramp capacitor are maintained in discharged states.

At the start of a measurement cycle, the Arduino code places S2 into the Charge position and charges C-Ramp with a constant current of 9.4mA. The resulting voltage at the inverting input of the comparator increases at a steady rate of 20mV/ms (ie, 20V/s).

After 480µs, S1 is switched to the charge position for 20µs, connecting

Electrolytic capacitor construction

In their most basic form, capacitors have two conductive plates (the anode and cathode) separated by an insulating material called the dielectric. There are three main types of electrolytic capacitors based on the material used for the anode and the associated dielectric used in their design: aluminium, niobium oxide and tantalum.

Capacitance is directly proportional to the total surface area of the plates but inversely proportional to the distance between the plates. Hence, the thinner the dielectric, the more efficient capacitors become.

Dielectrics have a high resistance; for low-value capacitors, examples include various polymers, mica, ceramics and even some liquids and gases, including air. In all three types of electrolytics, the anode consists of the primary material (aluminium, niobium oxide or tantalum) and the dielectric is a very thin layer of the respective oxide (pentoxide for niobium) deposited on the face of the anode.

This very thin dielectric must be in close contact with the cathode, which is the electrolyte's purpose. In essence, the electrolyte is the actual cathode, except that we also require a physical connection that allows the device to be soldered into a circuit. To ensure a high-quality coupling with low resistance, the electrolyte is a highly conductive liquid, gel or solid.

In aluminium electrolytic capacitors, an efficient way to ensure a high-quality coupling between the two is to sandwich a thin electrolyte-soaked sheet of paper between the dielectric and the cathode. Manganese dioxide is a solid electrolyte typically used in niobium and tantalum capacitors to connect the cathode to the dielectric.

a constant current source to the capacitor being tested. Depending on the range, the applied current is either 0.5mA, 5mA or 50mA. The test current pulse is kept very short to minimise charge build-up on the capacitor plates; we only want to measure the momentary pulse that develops across the capacitor's equivalent series resistance.

Courtesy of Ohm's law, we know that the magnitude of the resulting voltage pulse is directly proportional to the ESR of the capacitor. The resulting test pulse voltage is amplified by a factor of 20 and is fed into the non-inverting input of the comparator.

It compares the magnitude of the test pulse to the reference ramp voltage, and if the magnitude of the test pulse is greater than the latter, the comparator produces a 5V pulse at its output. The Arduino code increments a counter and then waits another 480µs before closing S1 again for 20µs to produce another test pulse.

Since the ramp voltage increases at a constant rate, it will eventually exceed the magnitude of the test pulses. The Arduino code detects the missing pulse and stops the measurement process, placing both S1 and S2 in the discharge position. The Arduino uses the total number of pulses and the test current to calculate the ESR figure and displays it on the LCD screen.

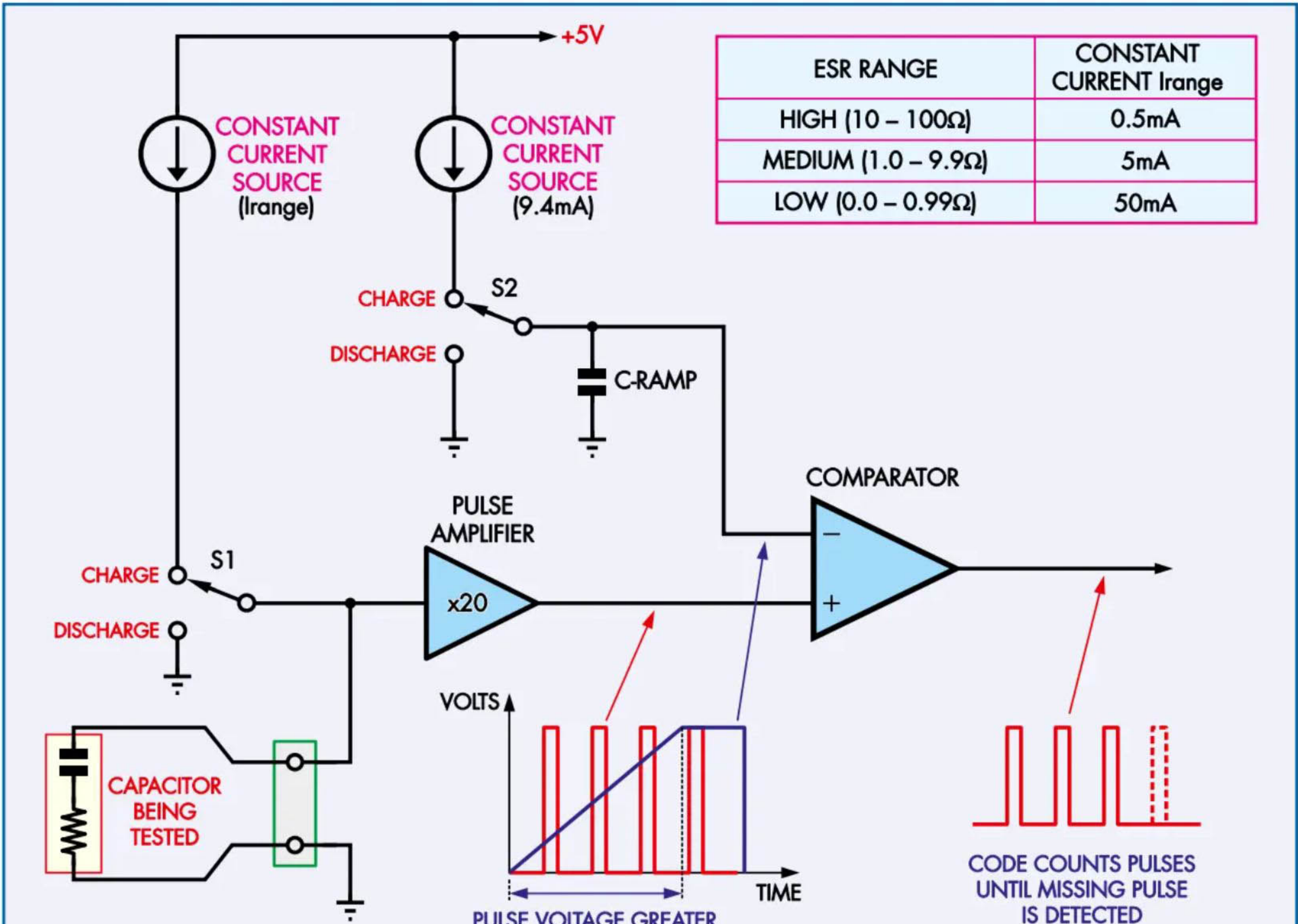
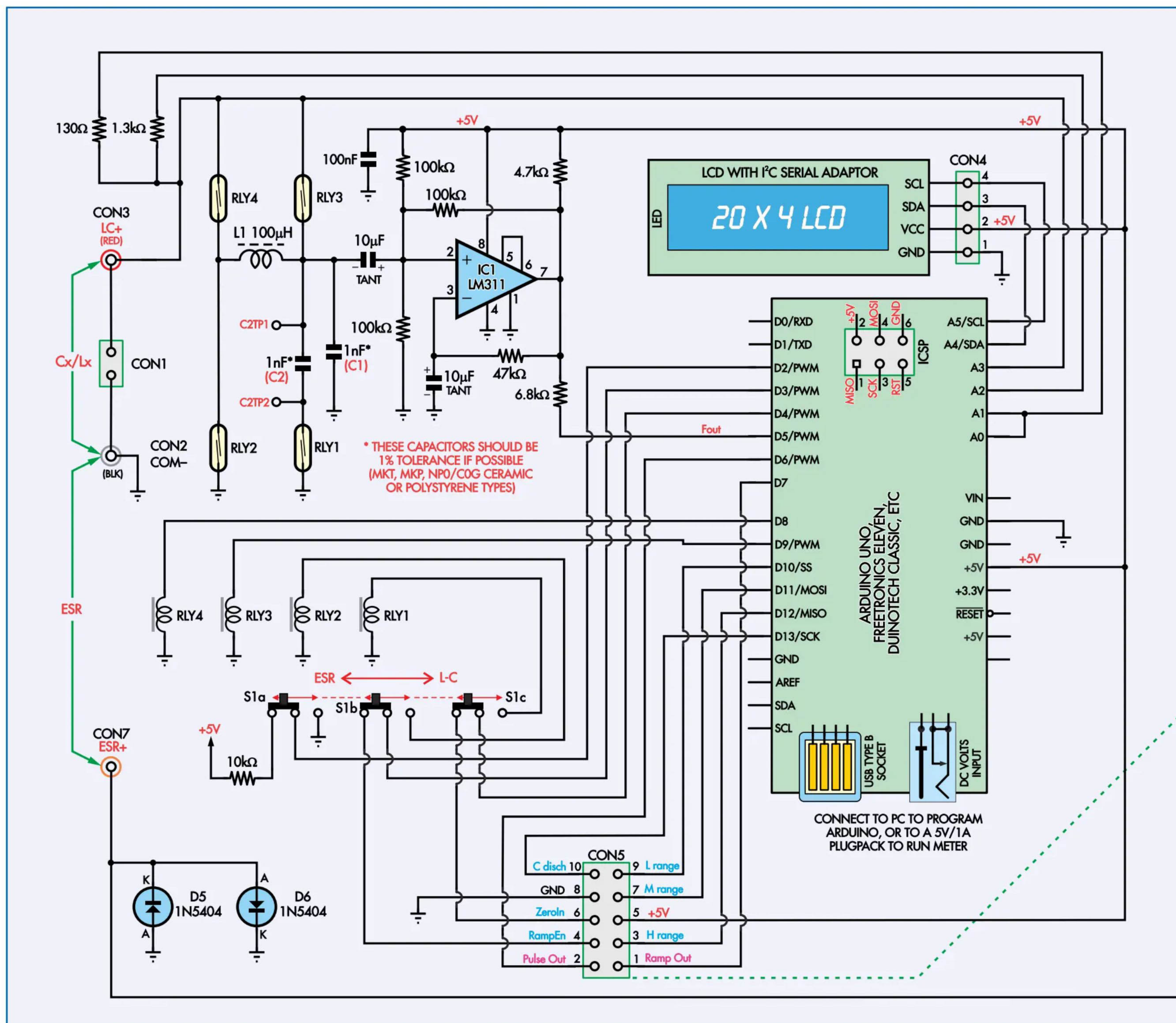


Fig.1: S1 repeatedly discharges and then briefly applies current to the DUT. The pulses are too short to charge the capacitor, so the resulting voltage is proportional to the ESR. The pulse amplifier then feeds an amplified version to the comparator, along with a linear ramp, and by counting the number of output pulses, we can accurately determine the ESR.

Circuit details

Fig.2 shows the circuit diagram of the original LC meter (on the left) with the ESR add-on on the right. However, note that some extra components are shown on the left, such as mode switch S1 and ESR input protection diodes D5 & D6.

While only one connection is shown passing between them – the



added ESR+ terminal connection – there are 10 further connections between the corresponding pins of CON5 and CON6. Note that GND is shared between both sides via pin 8 of those connectors.

We have produced two versions of the PCB. The smaller version that is an add-on to the existing LC Meter design only has the added circuitry on the right (with a few components mounted off-board, such as D5 & D6). However, the larger version of the PCB incorporates everything shown in Fig.2 and simplifies the wiring, especially since CON5 & CON6 are not required.

The ESR circuit on the right has three sections: 1) a set of current sources used to pulse the capacitor being tested (upper left), 2) the pulse amplifier (lower left) and 3) the reference ESR voltage ramp generator (upper right).

Pulse current sources

Transistors Q1, Q2 and Q3 are driven by Arduino Uno digital outputs D12, D11 and D10 when the respective output is pulled low. The Arduino Uno will switch on one of the transistors depending on the measurement range. The 10kΩ, 1kΩ & 100Ω collector resistors set the current pulse to 0.5mA, 5mA or 50mA.

There is no current regulation; we rely on the fact that the 5V supply is regulated, and the DUT is initially discharged when the current is applied. Therefore, close to 5V appears across the selected resistor and the current is determined by Ohm's law.

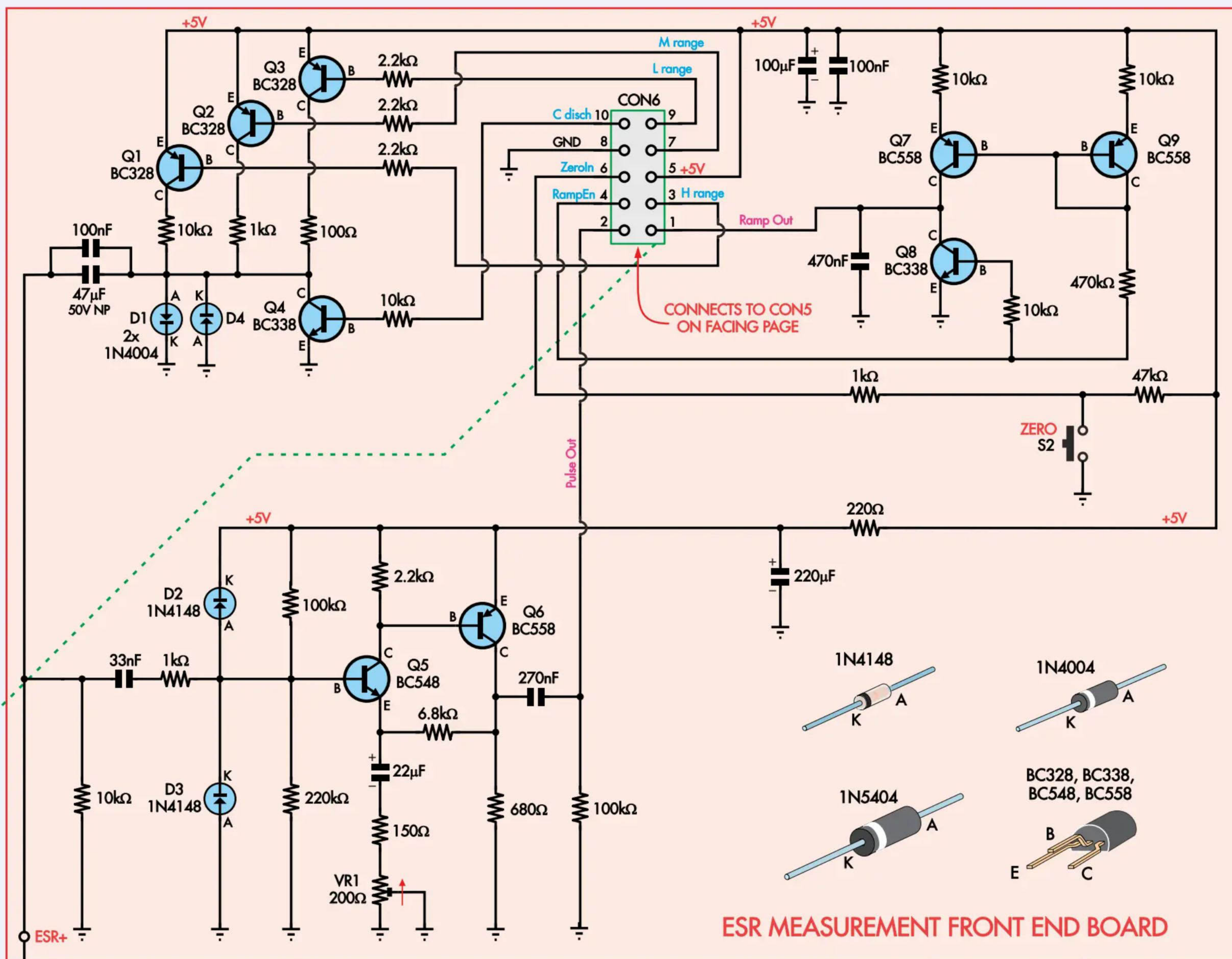
The current pulse is applied to the capacitor being tested via the parallel 100nF and 47μF capacitors which block any DC components. The ESR of this combination of capacitors is inconsequential, given

the relatively high values of the current source resistors. Critically, the measurement is taken directly from the DUT terminal, so the circuit is not measuring the ESR of those two capacitors as well.

The 100nF capacitor keeps the impedance low at high frequencies, as required by the nature of the short current pulses.

Whenever Q1, Q2 and Q3 are turned off, the Arduino Uno digital output D13 switches Q4 on by supplying current to its base. This ensures that the two AC-coupling capacitors are maintained in a discharged state, ready for the next current pulse.

Inverse parallel diodes D1 & D4 protect Q4 from potentially high currents should a charged capacitor be connected to the test leads. The maximum pulse voltage for an ESR value of 100Ω is typically under



Arduino-based L/C+ESR Meter

Fig.2: the original LC Meter circuit is on the left (with a few additions), while the added ESR-sensing circuitry is on the right. Headers CON5 and CON6 are not present on the combined PCB we've designed; instead, the ten connections are run via PCB tracks. Otherwise, a ribbon cable joins all pins between the two connectors.

500mV, so D1 and D4 have minimal effect on the pulse voltage.

Pulse amplifier

The pulse voltage developed across the capacitor being tested is fed to the pulse amplifier via a 33nF capacitor and a 1kΩ series capacitor. The pulse is amplified by a two-stage transistor amplifier formed by Q5 and Q6. The ratio of the 6.8kΩ feedback capacitor to the 150Ω fixed resistor and VR1 (adjusted for about 200Ω) sets the gain to 20 ($1 + 6.8\text{k}\Omega \div [150\Omega + 200\Omega]$).

Diodes D2 and D3 protect Q5 if a charged capacitor is connected to the test leads.

The amplified pulse voltage goes to the non-inverting input of the Arduino Uno's comparator via a 270nF capacitor, which blocks the DC voltage across the 680Ω resistor at Q6's collector. This resistor keeps

the 270nF capacitor discharged in the absence of a pulse.

Voltage ramp generator

PNP transistors Q7 and Q9 operate as a current mirror circuit to charge the 470nF ramp capacitor at a constant rate. When the Arduino pulls pin 4 of CON6 low, Q9 switches on, causing about 9.4μA to flow through the 470kΩ resistor. At the same time, Q8 switches off, allowing the ramp capacitor to charge. Q7 mirrors the current through Q9, so the capacitor begins to charge from 0V at 9.4μA.

The rising voltage across the 470nF capacitor is connected to the Arduino Uno's internal comparator (inverting input) via pin 1 of CON6. The Arduino Uno disables the ramp generator by setting pin 4 of CON6 high, turning off the charging via C9 while switching on Q8 to discharge the ramp capacitor.

Integration with the LC meter

The LC meter used the Arduino's analog comparator inputs (D6 and D7) as digital outputs to drive the coils of relays RLY1 and RLY2. It was necessary to move those functions to D3 and D4 (by modifying the LC Meter code) to allow the ESR function to use the comparator. The larger, combined PCB design includes this rerouting.

At the same time, D3 and D4 are shared with the ESR meter as digital I/Os via the selector switch, S1, that chooses between the LC and ESR modes. This was necessary since there were insufficient spare I/Os available on the Uno. As the original LC Meter shield lacks CON5, the wires from CON6 go to the Arduino/switch pins on my prototype.

Additional input protection

If the ESR meter were accidentally connected to a charged capacitor,

the energy dumped into this circuit could still damage it despite the protections mentioned above. As with a much earlier design, we have included two high-current diodes (1N5404s) connected back-to-back directly across the input sockets: D5 and D6. Despite this, remember to discharge capacitors before testing them!

Software

The ESR measurement code is based on Bob Parker's algorithm published

in Silicon Chip in the March 2004 issue, with minor changes to the pulse timings to better suit the Arduino Uno. The original design uses a pulse width of 8µs with an off-time of 492µs. Such settings resulted in a slight fluctuation in the readings.

For example, a 0.6Ω resistance would show a reading fluctuating between 0.59 and 0.61. A pulse width of about 20µs improved the stability with no impact on accuracy, so a 480µs off-time was adopted to maintain the overall 500µs period.

The program starts in the high range by setting D12 low and D13 high. This sets the pulse current to 0.5mA. At the same time, the reference voltage generator is initiated by setting D3 low.

If the pulse at the Arduino's non-inverting input exceeds the reference voltage, the comparator's interrupt-on-change feature sets a flag indicating a pulse was detected. Consequently, a counter is incremented, the interrupt flag is reset, and another pulse is applied to the capacitor being tested.

This process repeats until the code detects that the flag was not set after applying a current pulse. This signifies that the reference ramp voltage has reached a level greater than the pulse voltage, and therefore the counting is complete.

After each count cycle, if the total number of pulses is below 10, the next lower range is selected, and the measurement is repeated until the count produced is between 10 and 100. In the low range, a count between 10 and 100 equates to an ESR reading of between 0.1Ω and 1Ω; in the medium range, it represents 1Ω to 10Ω; or 10Ω to 100Ω in the high range.

If the count exceeds 100, the program automatically tries the next higher range until the count is between 10 and 100. If the count remains above 100 on the highest range, then the display will show "Over range!".

Test lead resistance

Since we aim to measure ESR values well below 1Ω, the resistance of the test leads and banana socket connections can introduce errors. Therefore, if the Zero button (S2) is pressed, the Arduino notices its D4 input pulled low and shows the message "Short test leads and press zero...".

Once the leads are shorted, the Arduino repeatedly measures and displays the lead resistance in ohms on the fourth line of the LCD. The code now waits for the zero button to be pressed again and saves the lead resistance in the Arduino's EEPROM. The stored result is then subtracted from the subsequent capacitor ESR measurements.

In addition to displaying ESR measurements on the LCD, the Arduino also produces a serial stream of the measurement data via its USB port.

The incremental count is displayed for each current pulse, followed by the final count, the final range selected and the number of

Parts List – Arduino ESR Meter

- 1 suitable case [Altronics H0401]
- 1 Arduino Uno or equivalent microcontroller module
- 1 20×4 blue backlit alphanumeric LCD with I²C interface [SC4203]
- 1 double-sided PCB coded 04106182, 68.5 × 115.5mm
- 1 100µH bobbin-style or high-current axial RF inductor (L1)
- 4 5V DC coil DIL reed relays (RLY1-RLY4) [Altronics S4100, Jaycar SY4030]
- 1 200Ω top-adjust multi-turn trimpot (VR1)
- 1 3PDT solder tag toggle switch (S1) [Jaycar ST0505]
- 1 vertical tactile pushbutton switch (S2)
- 3 PCB-mount right-angle banana sockets; one black, two red (CON2, CON3, CON7) [SILICON CHIP SC4983] OR
- 3 panel-mount banana sockets; two black, one red (CON2, CON3, CON7)
- 1 4-pin right-angle polarised header with matching plug and pins (CON4)
- 1 set of Arduino-style regular headers (1×10-pin, 2×8-pin, 1×6-pin)
- 1 100mm length of 4-way ribbon cable terminated with DuPont sockets at one end
- 8 M3-tapped 12mm spacers
- 9 M3 × 6mm panhead machine screws
- 4 M3 × 6mm countersunk head blackened machine screws

Semiconductors

- 1 LM311 high-speed comparator, DIP-8 (IC1) [Altronics Z2516, Jaycar ZL3311]
- 3 BC327 or BC328 500mA PNP transistors (Q1-Q3)
- 2 BC337 or BC338 500mA NPN transistors (Q4, Q8)
- 1 BC548 or BC547 100mA NPN transistor (Q5)
- 3 BC558 or BC557 100mA PNP transistors (Q6, Q7, Q9)
- 2 1N4004 400V 1A diodes (D1, D4)
- 2 1N4148 75V 200mA diodes (D2, D3)
- 2 1N5404 400V 3A diodes (D5, D6)

Capacitors

- 1 220µF 16V electrolytic
- 1 100µF 16V electrolytic
- 1 47µF 16V non-polarised electrolytic
- 1 22µF 16V electrolytic
- 2 10µF 6.3V tantalum or ceramic
- 1 470nF 63V MKT
- 1 270nF 63V MKT
- 3 100nF 50V multi-layer ceramic or MKT
- 1 33nF 63V MKT
- 2 1nF 1% NP0/C0G ceramic, MKP or polystyrene [SILICON CHIP SC4273]

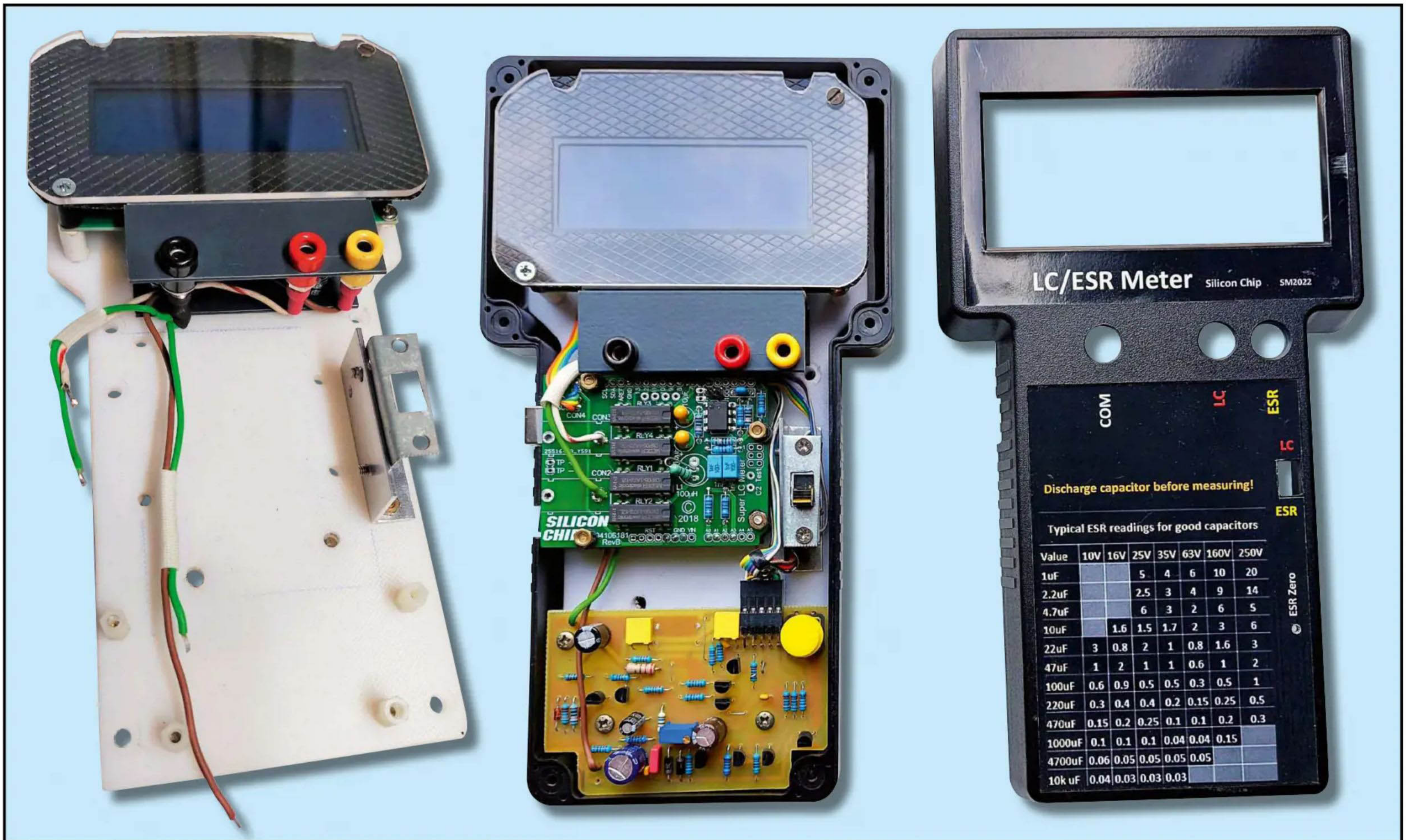
Resistors (all 1/4W 1% axial)

- | | | | | |
|---------|---------|---------|---------|--------|
| 1 470kΩ | 1 220kΩ | 5 100kΩ | 2 47kΩ | 7 10kΩ |
| 2 6.8kΩ | 1 4.7kΩ | 4 2.2kΩ | 1 1.3kΩ | 3 1kΩ |
| 1 680Ω | 1 220Ω | 1 150Ω | 1 130Ω | 1 100Ω |

Extra parts if building the ESR Meter with separate PCBs

- 1 double-sided PCB coded 04106181, 68.5 × 53mm
- 1 3PDT solder tag slide switch (S1) [Mouser 502-50209LX]
- 1 2x5 IDC header with matching socket (CON6)
- Ribbon cable and heatshrink tubing

Value	4-band code	5-band code
100Ω		
130Ω		
150Ω		
220Ω		
680Ω		
1.0kΩ		
1.3kΩ		
2.2kΩ		
4.7kΩ		
6.8kΩ		
10kΩ		
47kΩ		
100kΩ		
220kΩ		
470kΩ		



The prototype Meter was made using a specialised case to suit the display module. While you can use multiple PCBs as shown above, a single board design using the PCB shown in the lead photo requires much less wiring.

range changes made during a measurement. Note that the accumulated count includes the effects of test lead resistances.

Combined LC / ESR Meter
When the LC Meter and ESR Meter are combined, a contact on the LC/ESR selector switch, S1, signals

which mode has been selected to the Uno via its digital input D2. With D2 low, it is in ESR mode. Switching from one mode to the other happens

What is a normal ESR value?

Electrolytic capacitors include reactive elements, so the ESR value will change depending on the frequency of the applied voltage (there is also an equivalent series inductance or ESL). Temperature changes also affect the reading, as do different manufacturing processes. Manufacturer data sheets typically give the expected ESR values at 20°C and 100Hz, 120Hz or 100kHz, although many do not include such information (or give it differently, eg as a dissipation factor).

Thus, providing definitive expected ESR values for all electrolytic capacitors is impossible. Still, we do not expect to see the values exceeding several ohms, and higher-value capacitors should generally have lower ESR values. Capacitors designed for use in switch-mode supplies (often labelled “Low ESR”) should have values of a fraction of an ohm or less.

For example, the data sheet for the Panasonic FM-A series of aluminium capacitors gives values from 0.012Ω to 0.34Ω ohms varying with the voltage rating (6.3V to 50V) and capacitance (22μF to 6800μF). The data

sheet for the RubyCon YXF series for similar capacitance and voltage ranges lists the maximum expected ESR values to be between 0.025Ω and 1.3Ω.

Table 1 below shows tabulated typical ESR values. These are

generalised expected readings, so manufacturer data sheets should be used as a reference. However, it should be apparent that a capacitor is faulty if the measured ESR value exceeds tens or even hundreds of ohms!

Table 1: typical ESR readings for good capacitors

	10V	16V	25V	35V	63V	160V	250V
1μF			5	4	6	10	20
2.2μF			2.5	3	4	9	14
4.7μF			6	3	2	6	5
10μF		1.6	1.5	1.7	2	3	6
22μF	3	0.8	2	1	0.8	1.6	3
47μF	1	2	1	1	0.6	1	2
100μF	0.6	0.9	0.5	0.5	0.3	0.5	1
220μF	0.3	0.4	0.4	0.2	0.15	0.25	0.5
470μF	0.15	0.2	0.25	0.1	0.1	0.2	0.3
1000μF	0.1	0.1	0.1	0.04	0.04	0.15	
4700μF	0.06	0.05	0.05	0.05	0.05		
10mF	0.04	0.03	0.03	0.03			

A version of Table 1 that can be downloaded as a PDF.

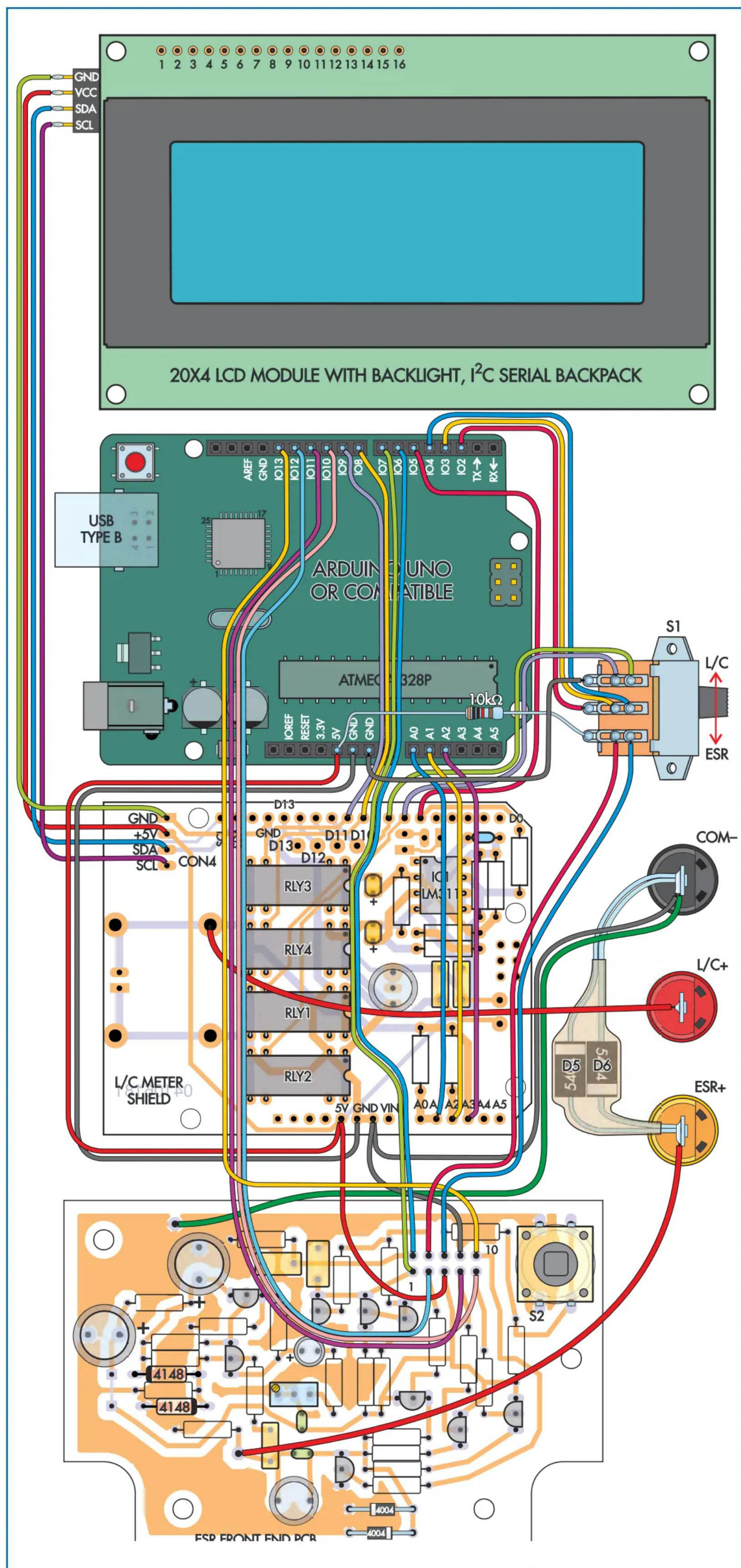
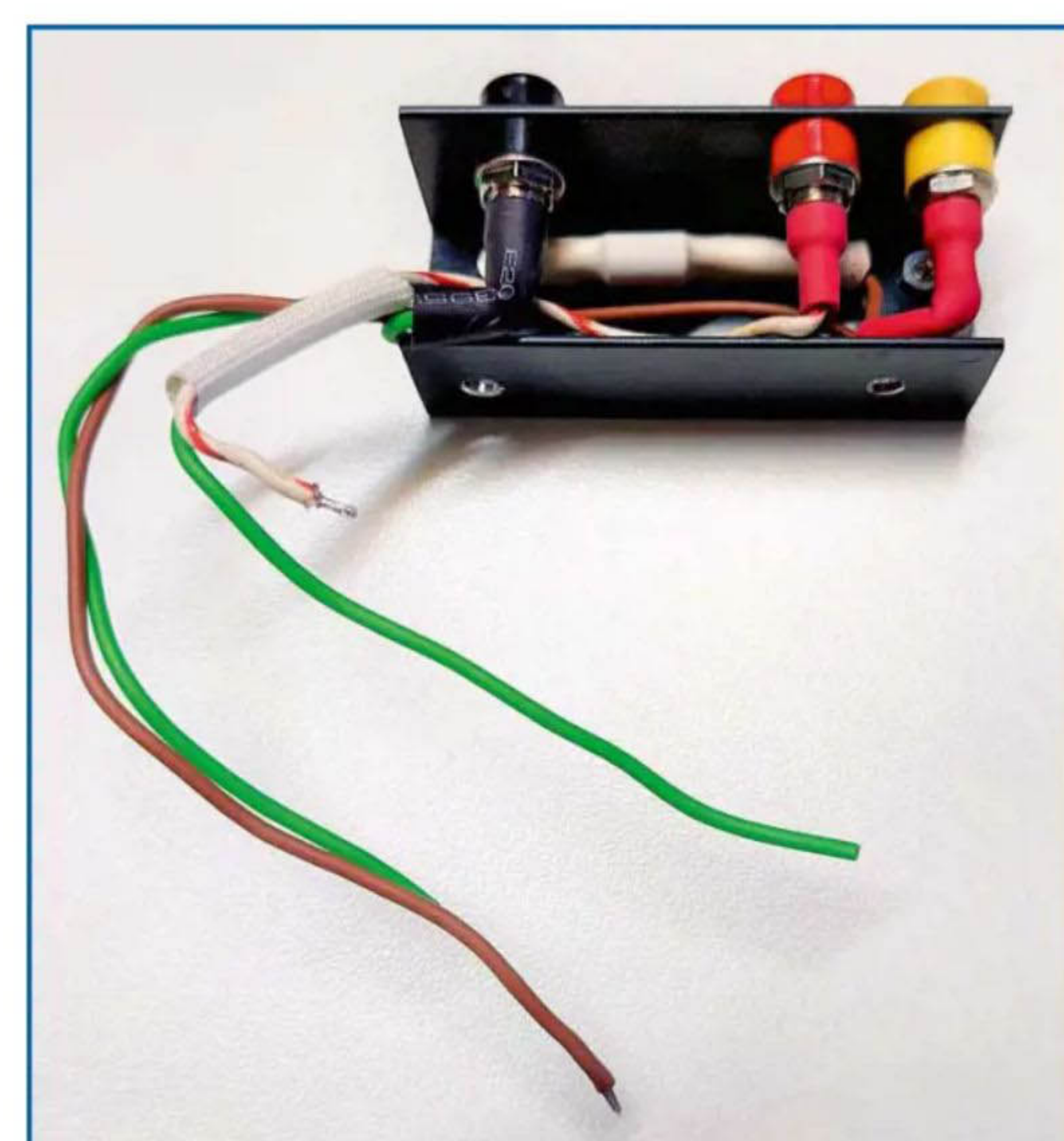


Fig.3: this is the wiring needed to add the ESR feature to the existing LC Meter design by simply adding another small board (at the bottom). We think most constructors will prefer the much easier method of building the single combined PCB!



If you decide to build the ESR meter as separate PCBs, you might also need a mounting arrangement for the banana sockets as shown here and in Fig.8. In this case diodes D5 & D6 are located inside the white heatshrink.

after the program completes the current procedure being processed by the Arduino.

As previously mentioned, Arduino pins D3 and D4 are shared between the LC and ESR modules. D3 serves as a digital output in both modes; however, D4 is a digital input for the ESR module (for the Zero switch) but an output for the LC meter (driving RLY1). When switching modes, D4 is reconfigured by the code as required.

Case selection

The case used for the prototype is available from Mouser Electronics ([563-HH-3421](#)) or Digi-Key ([HH-3421-ND](#)), although stocks are limited. An optional tilt stand is available separately from Digi-Key ([377-1171-ND](#)).

Because the combined board is considerably narrower than the ESR-only board, it should fit in that case. With an internal depth of 37mm (excluding things like mounting bushes, which could be removed), the Arduino and control board stack should fit, as should mode switch S1, but it will be a bit of a squeeze.

Alternatively, you could use just about any rectangular case. It would need to be at least 175mm tall internally for a 20x4 LCD module to fit at the top with the combined control PCB and Arduino below it. The LCD will be around 87mm wide, defining the minimum internal width, while a depth of at least 30mm is required to fit the Arduino Uno, the shield on top of it, and the body of switch S1.

The Altronics H0401 sloped case specified in the parts list should have plenty of room. Because of the

(watch their orientation), diodes D5 & D6 (ditto), the banana sockets and inductor L1.

That just leaves the headers and the 3PDT mode switch. The CON4 header needs to be fitted as we'll use it to connect to the LCD later, unless you plan to solder the LCD wires directly to its pads. CON1 is only needed if you plan to mount the banana sockets off-board and will not solder the wires directly (although you will need to do so for CON7 regardless).

The remaining headers mount on the underside of the board. Use standard pin headers for the four SIL connections to the Arduino Uno (or similar) since we will not stack anything on top of this board. However, they need to be fitted using a particular method due to the height of the USB connector on the Arduino Uno board that will fit below.

First, apply some insulation to the top of the USB socket on the Uno, such as electrical tape or Kapton tape. Next, insert the Arduino

headers into the shield board from the underside. Place a scrap of perfboard, protoboard or similar on top of the header pins that stick out the top of the board, then use a flat object to push the headers down so the tops of the pins are flush with the perfboard.

Carefully remove the perfboard without moving the headers, then solder the pins at either end. This will mean there is a gap between the underside of the PCB and the plastic spacer on the headers. That's so the pins project out further to reach the Arduino sockets despite the USB socket not allowing the shield to be pushed fully down.

Finally, the 3PDT toggle switch mounts on the top side of the board into slotted holes designed to suit its rectangular solder lugs. This avoids the need to run nine flying leads, although you could do so if you want to mount that switch elsewhere. If doing so, use a short length of ribbon cable.

Testing

Make a final inspection of the soldering to ensure there are no solder bridges between tracks and that all the components are in their correct position and correctly orientated.

If you have built the separate ESR board, you can do some testing before you wire it up. Connect pin 5 of CON6 to a +5V supply with pin 8 at 0V.

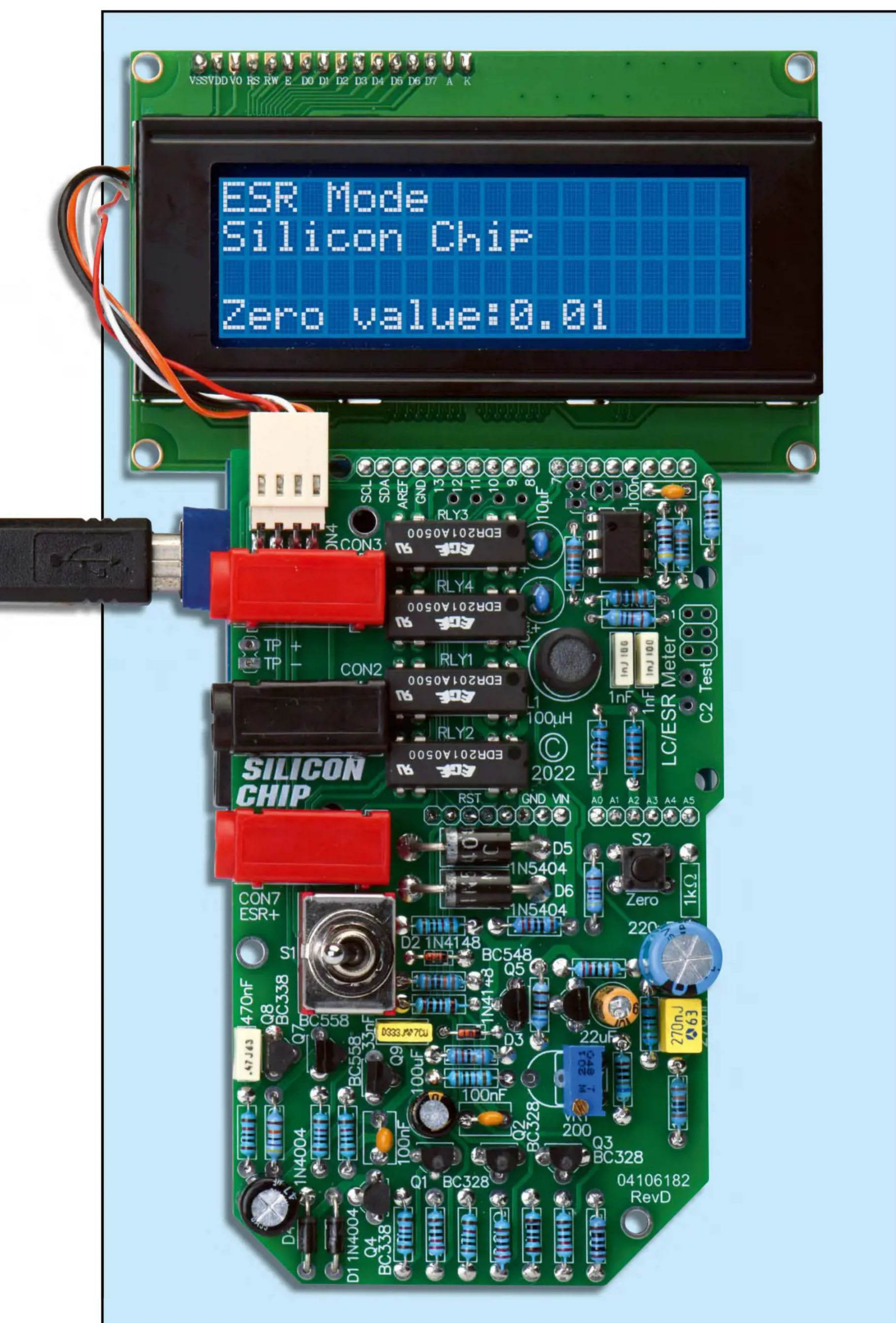
Measure the current draw, which should be about 1mA. If the current is significantly higher (or zero), disconnect the supply and look for assembly errors.

When plugging the shield into the Arduino, we recommend that you use 12mm tapped spacers and short machine screws to hold the two boards together due to the fact the headers won't plug fully into the sockets. Attach the four spacers to the mounting holes on the Arduino, but only one needs to be screwed in through the shield to hold it down. The rest just set it at the correct height.

If there is a solder joint touching the top of the USB socket that prevents you from tightening the screws, you should trim it flush to the extent possible.

Wiring

If you are building the combined PCB, there isn't much to the wiring. You just need to make up a 4-way cable to go from CON4 to the I²C LCD header. Make sure the connections are made per the labelling on the two



When the Meter is switched to ESR mode, a splash screen is briefly displayed showing the 'Zero value', which is effectively the offset due to the resistance of the leads and anything else that might be in the measuring circuit.

PCBs, ie, GND to GND, SDA to SDA etc. If in doubt, refer to Fig.3; using a 4-way ribbon cable will keep it tidy. If you haven't already soldered the I²C adaptor to the LCD screen, do that now, as the 4-way cable from the main board connects to that.

If you're adding the ESR board to an existing LC Meter, or building the boards separately for some other reason, wire them up as per Fig.3. The ten wires from CON6 are shown separately for clarity but again, it's best to use a 10-way ribbon cable and only split out the individual wires as much as necessary to reach the appropriate pads.

Note how, in Fig.3, the LC Meter shield no longer plugs directly into the Arduino since many of the pins are rerouted. Also note that diodes D5 & D6 are mounted off-board in this case.

Loading the software

To upload the firmware for the Uno board, you need to have the Arduino IDE (Integrated Development Environment) software installed on your computer. If you don't have it, get it from: www.arduino.cc/en/main/software

The program that runs on the Uno requires an external library to interface with the I²C LCD. Open the IDE and select: Sketch → Include Library → Manage Libraries... ,then search for "liquidcrystal_pcf8574" and install the version by Matthias Hertel.

Now open the sketch file: "ESR.ino" for the standalone version or "LC_ESR_Meter.ino" for the combined version. Select the board type as Arduino Uno (Tools → Board Type → Arduino AVR Boards), then use the Tools → Port menu to select the serial port that the Arduino is plugged into. Most versions of the Uno will display as COMx: (Arduino Uno or similar) in the dropdown menu.

If you're using a 16×4 LCD rather than the 20×4 LCD recommended, change the line "lcd.begin(20, 4)" to "lcd.begin(16, 4)". Compile and upload the sketch by pressing Ctrl-U. If you see the message "Done Uploading" at the bottom of the window, then all is well. If you get an error message, check that the LCD I²C library is installed correctly and that the correct serial port is selected.

LCD adjustment

If the LCD backlight is not lit, check that the backlight jumper is fitted on the I²C adaptor board. If the backlight is working, but there is no text, adjust the contrast pot on the back of the I²C adapter board.

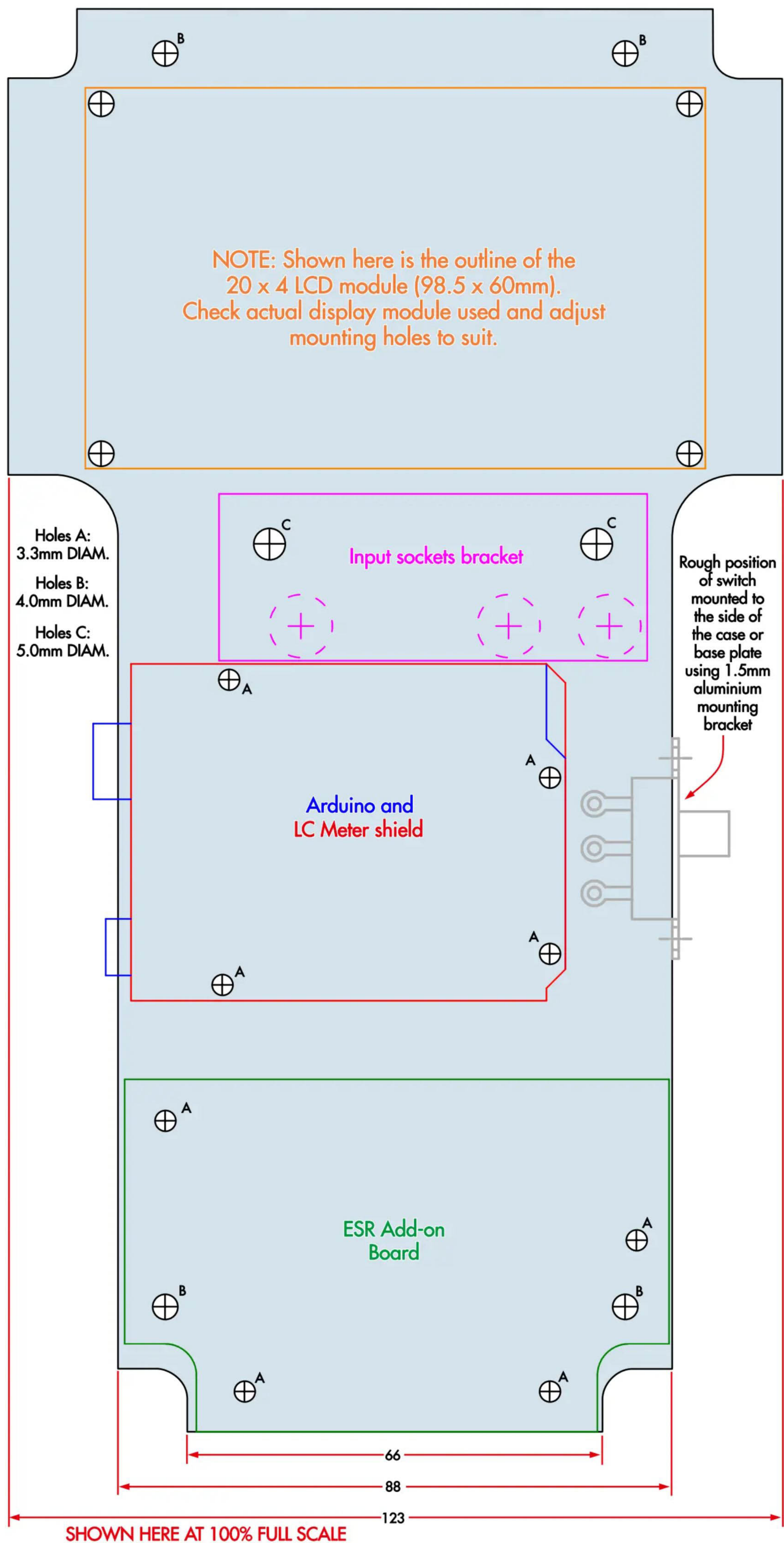


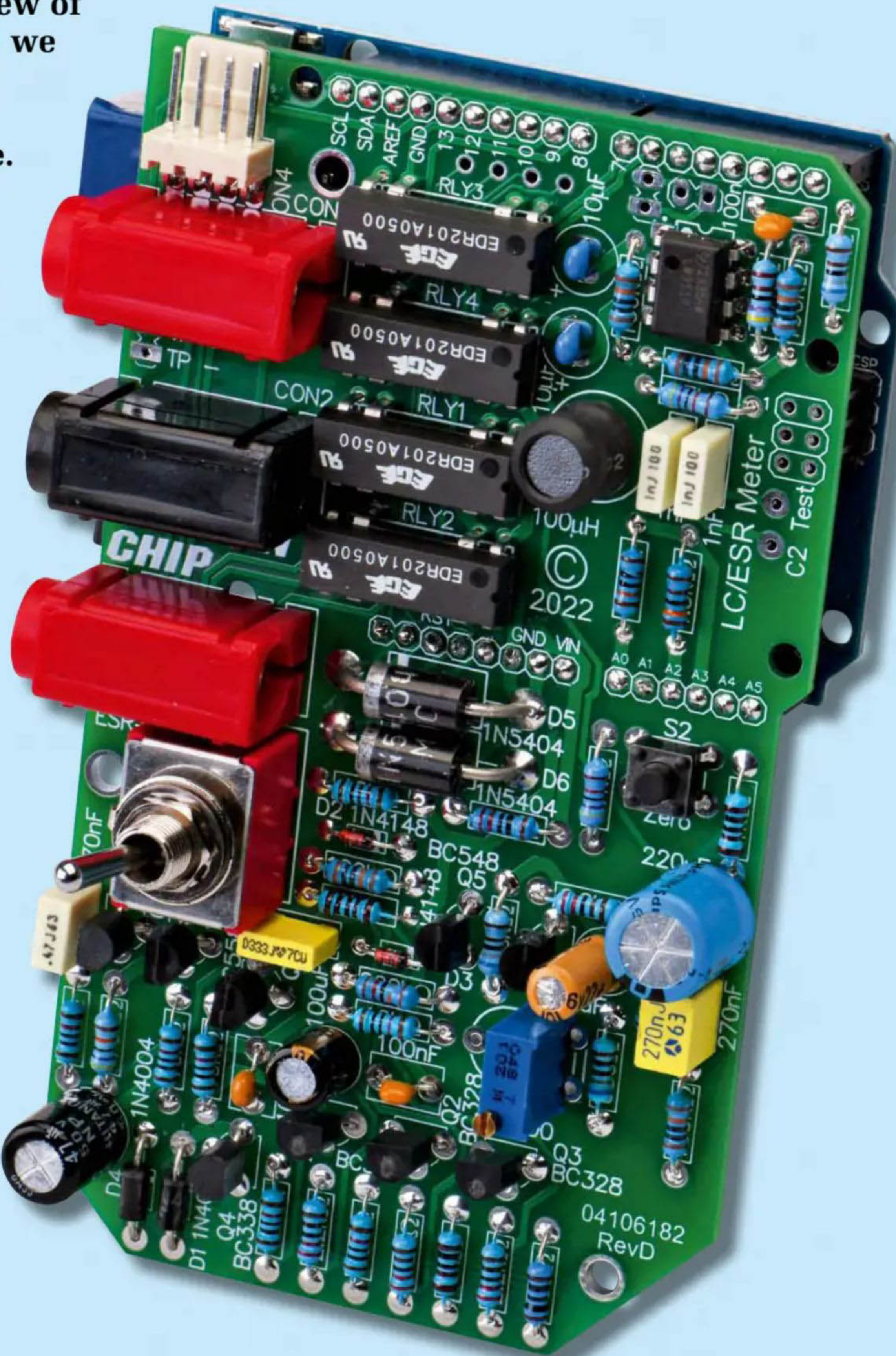
Fig.7: while not recommended for the combined PCB, here is how the separate PCBs were mounted on an acrylic baseplate for the prototype.

Zeroing the test leads

The program first checks to see if the resistance of the test leads

has been saved in the EEPROM; if not, you will be prompted to perform the Zero process. Follow the

Here is another view of the combined PCB we designed, plugged into an Arduino Uno, at actual size. Note how as well as the banana sockets projecting off the left-hand side, the USB and DC power inputs of the Uno do too. This allows you to make holes in the side of the case for all of those connectors.



the total resistance of the leads to be less than 1Ω or it won't accept the result and briefly display the message "Invalid reading or bad leads" before aborting the zeroing process.

In normal measurement mode and with the test leads separated, the display should indicate "Over range".

Calibration

Calibration is straightforward, using a known resistance of about 68Ω or 82Ω . Verify the resistor's actual value beforehand with a multimeter (deducting the multimeter lead resistances measured when shorting the leads together). Switch S1 (if present) to ESR mode. With this resistor connected via the probes, the screen should display a value close to the resistor value.

Adjust VR1 until the reading matches the resistor value. Now try a resistor in the medium range ($1-9.9\Omega$) across the leads and verify that the reading is close to expected. Similarly, a $0.1-0.9\Omega$ resistor should give a very close measured result.

With calibration complete, you can now test a selection of electrolytic capacitors to get a feel for the meter's operation.

The screen shows the measured ESR on the first line, the range (High, Medium or Low) on the third line, and the saved lead resistance on the fourth line. There is no need to subtract the lead resistance from the displayed ESR value, as that has already been done.

If you've built the combined LC/ESR Meter (as we think most people will), now would also be a good time to switch over to LC mode and verify that the unit changes modes when you flick the switch and that inductor and capacitor measurements are correct.

Final assembly

If you are building the unit as a standalone ESR meter, all that remains is to place the Arduino and ESR shield into an appropriate enclosure, with the LCD visible and the test lead terminals (and possibly S2) accessible.

We have not shown the wiring to achieve this but it is similar to what is shown in Fig.3 without the LC Meter shield. The main differences are that the two connections from pins D3 & D4 on the Arduino to the ESR PCB via S1 should be run directly, while pin D2 should be tied to GND. The 5V and GND supply connections also connect from the Arduino to the ESR board rather than via the LC Meter Shield.

```
ESR: Over range!
Range: High
Leads: 0.01 ohms
```

Screen 1: This screen is seen when no component is connected, or when a resistance over 100Ω is detected. The bottom line continues to display the lead resistance. If you see this when a capacitor is connected, it's probably not good anymore!

```
Short test leads
and Press ZERO..
0.01
```

Screen 2: Pressing and holding the ZERO button brings up this screen. You should short the ESR measurement terminals using the leads you would use for measurement and confirm that a low value as seen is displayed before pressing the ZERO button again.

instructions requesting the test leads to be shorted, and once the displayed resistance is stable, press the zero switch (S2).

The display should briefly indicate that the zeroing process is complete before changing to the regular measurement display. The code expects

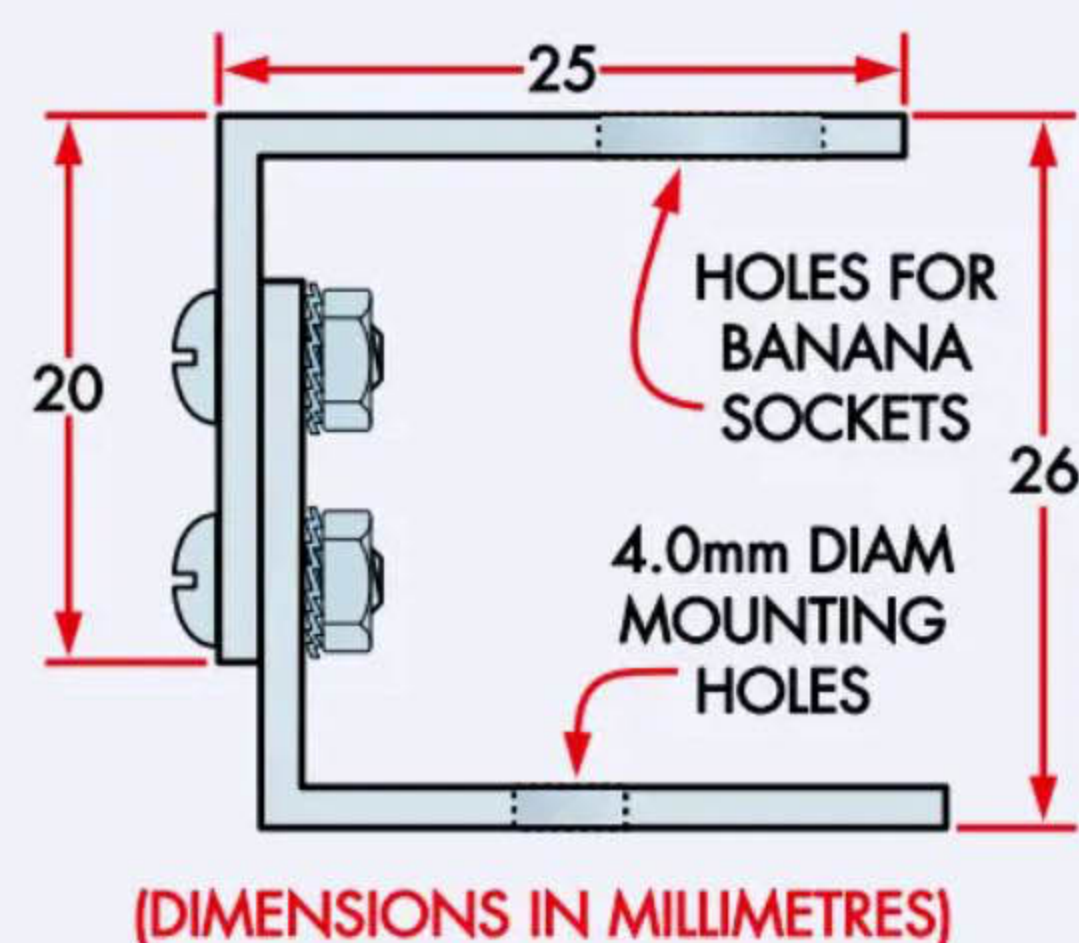


Fig.8: while also not necessary for the version built with the combined PCB as described, this shows the mounting bracket used to hold the banana sockets in the prototype.

If you have built the combined PCB, fitting it into an enclosure is a bit more straightforward. Again you will need a cut-out to view the LCD screen (unless your case has a clear lid) and possibly a way to access S2 (eg, a small hole in the case).

The toggle switch will fit through a hole in the lid of your enclosure, but the board should be mounted against the left edge so that the banana sockets will fit through holes in the side (unless you've decided to mount them elsewhere and connect them to the socket pads via flying leads). You could use panel-mount banana sockets mounted just off the

left edge of the board and attached via short wires.

As in the prototype, you would typically mount the LCD screen near the top of the case with the main PCB below.

Power for the prototype was fed in via the Uno's power connector, with the plug going through a cut-out in the left-hand side of the case. You could use a similar arrangement, or use a chassis-mounting DC barrel socket mounted elsewhere and wired to the VIN and GND pins of the Uno.

If your enclosure doesn't have a clear lid, cut a piece of clear acrylic or other plastic for the display window. You can either glue this onto the underside of the enclosure cover or mount it on top of the LCD.

Prototype mounting details

Some constructors may wish to use a similar mounting arrangement to the prototype. However, this is not suitable if you are using the combined PCB; it's only relevant if you have separate PCBs. The boards, mounting brackets and display were mounted on a 4mm-thick acrylic base plate, as shown in Fig.7.

Fig.8 shows the reinforcement bracket used for mounting the banana sockets to the case. Depending on

your case, you may not need this; you can mount the sockets directly to it, or use the onboard ones.

If you are chassis-mounting mode switch S1, you might want to make a similar bracket for it if mounting it directly to the case isn't suitable.

Consider that you should drill a hole about 3mm in diameter in the case for accessing Zero switch S2 later should you need to recalibrate the lead resistance. Having to open up the case to do that could be a nuisance.

Table 1 can be printed onto adhesive paper or printed, laminated and glued onto the case as a guide. That is what I did for the prototype. Keep in mind that if you're using the Combined PCB, the mode switch toggle will be in the middle of where I attached it on my prototype. Depending on where you've put the banana sockets, you may be able to attach it higher up to clear that switch.

Conclusion

Don't forget to discharge the capacitors before testing them!

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au

**GET THE
LATEST COPY
OF OUR
TEACH-IN SERIES
AVAILABLE
NOW!**

Order direct from Electron Publishing

PRICE £8.99
(includes P&P to UK if ordered direct from us)

Teach-In 9 – Get Testing!

This series of articles provides a broad-based introduction to choosing and using a wide range of test gear, how to get the best out of each item and the pitfalls to avoid. It provides hints and tips on using, and – just as importantly – interpreting the results that you get. The series deals with familiar test gear as well as equipment designed for more specialised applications.

The articles have been designed to have the broadest possible appeal and are applicable to all branches of electronics. The series crosses the boundaries of analogue and digital electronics with applications that span the full range of electronics – from a single-stage transistor amplifier to the most sophisticated microcontroller system. There really is something for everyone!

Each part includes a simple but useful practical test gear project that will build into a handy gadget that will either extend the features, ranges and usability of an existing item of test equipment or that will serve as a stand-alone instrument. We've kept the cost of these projects as low as possible, and most of them can be built for less than £10 (including components, enclosure and circuit board).

PLUS! You will receive the software for the *PIC n' Mix* series of articles and the full *Teach-In 2* book – *Using PIC Microcontrollers – A practical introduction* – in PDF format. Also included are Microchip's *MPLAB ICD 4 In-Circuit Debugger User's Guide*; *MPLAB PICkit 4 In-Circuit Debugger Quick Start Guide*; and *MPLAB PICkit4 Debugger User's Guide*.



ORDER YOUR COPY TODAY: www.electronpublishing.com

WHERE THE PICO MITE MEETS THE WEB

THE WEBMITE

A RASPBERRY PI PICO WITH MMBASIC, WIFI AND INTERNET CONNECTIVITY

| Article and MMBasic by Geoff Graham | WebMite firmware by Peter Mather |

Most readers will be familiar with the PicoMite, which we introduced in the July 2022 *Make it with Micromite* column. It is a Raspberry Pi Pico programmed in the MMBasic language and is a very capable microcontroller at a low price.

Following that, the Raspberry Pi Foundation released the Raspberry Pi Pico W, which is very similar to the original Pico but with the addition of a WiFi module.

In theory, you could take any Pico project and then add an internet connection... but it is not quite that easy. The internet and its protocols are complicated, with many protocol layers. That means it takes an experienced programmer to accomplish even the simplest task.

That is where our new WebMite comes in. We have added to the PicoMite firmware support for the wireless capability of the Pico W and the protocols (802.11n, DHCP, WPA-PSK, TCP, IP, TLS, HTML etc) that are necessary to access the internet. With the WebMite, you can easily:

- ☑ Connect to a WiFi network with a specified SSID and password.
- ☑ Implement a web server with advanced features.
- ☑ Query websites for data.
- ☑ Get the current time/date.
- ☑ Check the weather.
- ☑ Send emails.
- ☑ Publish/retrieve data from MQTT broker services (for IoT messaging).

- ☑ Remotely edit BASIC programs.
- ☑ Transfer files to and from the Pico over WiFi using TFTP.

These features have been implemented using an easy-to-use programming framework with the flexibility to handle the unusual aspects of accessing internet resources. Later in this article, we will present an example of a simple web server that uses just 12 lines of BASIC code – that is all it takes to serve up a web page for your projects.

The MMBasic language is an easy-to-use programming environment. With the WebMite, you can easily add internet features to complex gadgets with touch-sensitive LCD panels, SD card support for storing files, connection to various sensors and so on.

Loading the firmware

As the hardware is already built for you (the Raspberry Pi Pico W, available for under \$10), all you have to do to create the WebMite is load the appropriate firmware onto that module. Luckily, that is easy.

The WebMite firmware can be downloaded for free from the August 2024 page of the *PE* website: <https://bit.ly/pe-downloads> or the author's website at <http://geoffg.net/webmite.html>

It comes with a comprehensive 178-page user manual that includes a tutorial on BASIC programming for programming beginners.

Loading the WebMite firmware on the Raspberry Pi Pico W is the same as with the original Raspberry Pi Pico and is described in the user manual. Essentially, you plug your Pico W into a USB port on your computer while holding down the button on the top of the module. The Pico will then create a pseudo USB drive on your computer, and you just drag and drop the WebMite firmware into that.

When the transfer has been completed, the Pico will restart running the WebMite firmware and create a serial connection via its USB port.

Once it has done that, you can use a terminal emulator like Tera Term (<http://tera-term.en.lo4d.com>) to access the WebMite's console. If you then hit the Enter key on your keyboard, you will see the MMBasic prompt, which is just a greater-than character (>). You can configure the WebMite, test commands, edit programs, and run them at this command prompt.

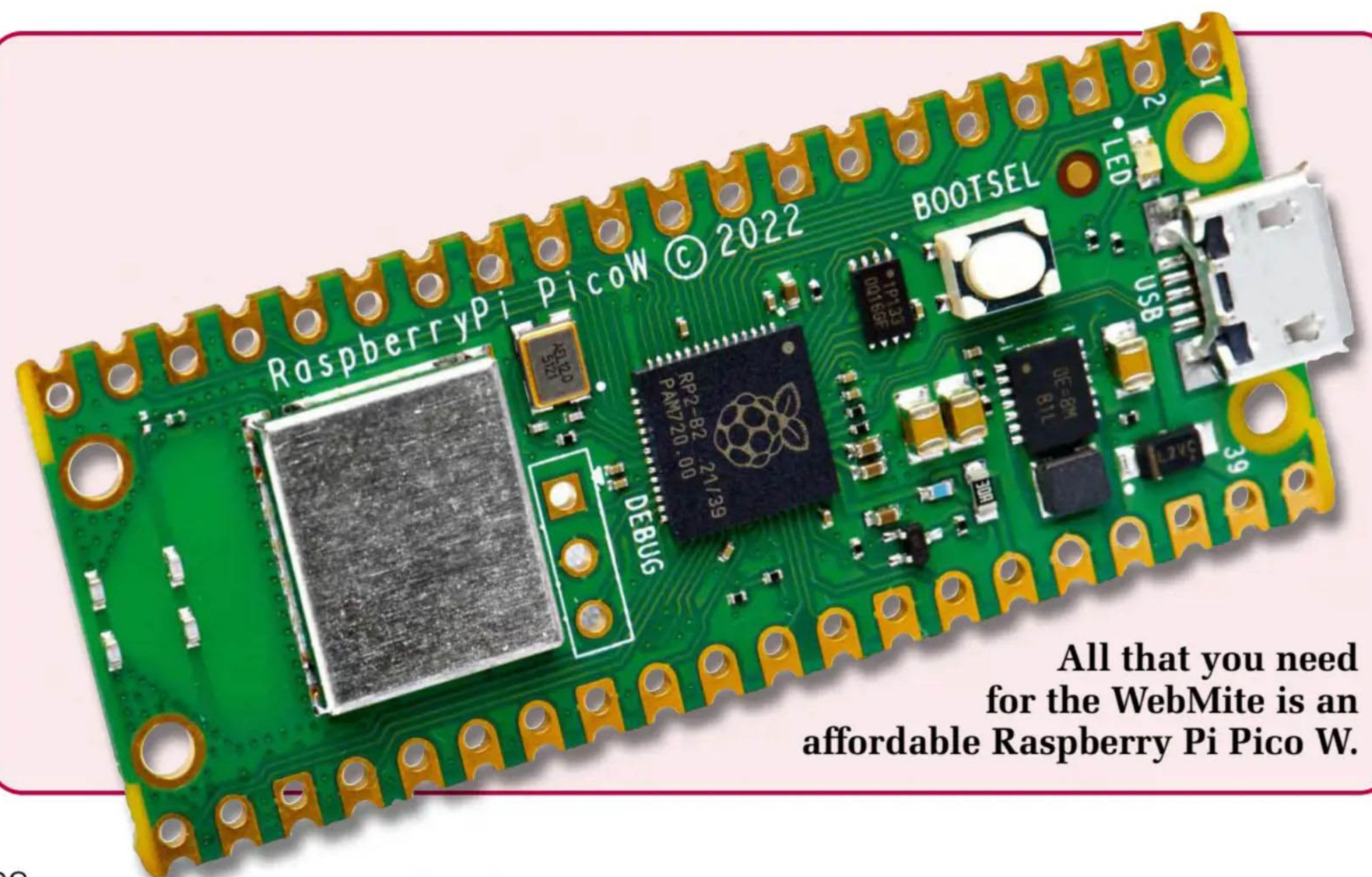
Internal file system

Before we get into the internet capabilities of the WebMite, we need to introduce the internal file system. It looks like an SD card to the programmer, but files are actually stored in the flash memory chip of the Raspberry Pi Pico W.

You can open files for reading and writing, create and navigate directories and do all the things you can do with a physically attached SD card. That includes using the normal BASIC file commands such as OPEN, CLOSE, FILES etc. If you connect a removable SD card to the WebMite, it is accessed as drive "B:" while the internal file system is drive "A:".

This internal file system has a capacity of about 600kB, and it is automatically created by the firmware when MMBasic is loaded onto the Raspberry Pi Pico (W). This is especially useful on the WebMite because, to set up a web server, you need somewhere to store the web pages and images to serve. The internal file system is perfect for the job.

This feature was introduced in the last release of the PicoMite firmware,



All that you need for the WebMite is an affordable Raspberry Pi Pico W.

so if you are currently using the PicoMite or the VGA PicoMite, consider upgrading to get this feature.

Connecting to WiFi

The WebMite can connect to any WiFi network running 802.11n (2.4GHz) with WPA-PSK security. The encryption must be either TKIP or AES (or both) and DHCP must be enabled on your router. These are standard requirements for most WiFi-enabled gadgets, so most routers are set up like this by default.

To log into your network, use the command `OPTION WIFI` at the WebMite's command prompt. For example, if your network is called MyNetwork and the password is secret, you would use this command at the command prompt (the quote characters are required):

```
OPTION WIFI "MyNetwork", "secret"
```

This will be remembered and will be automatically reapplied on every reboot. It will also cause the WebMite to restart and drop the USB connection, so you will have to reconnect to access the command prompt again.

You can check the IP address that your router gave to the WebMite with the command:

```
PRINT MM.INFO(IP ADDRESS)
```

Most routers will allocate the same address to the WebMite on every reboot. However, if you want to ensure the address will not change, go into your router's configuration and allocate a static IP address to the WebMite. Incidentally, you can have many WebMites on your network with different IP addresses and they will not conflict.

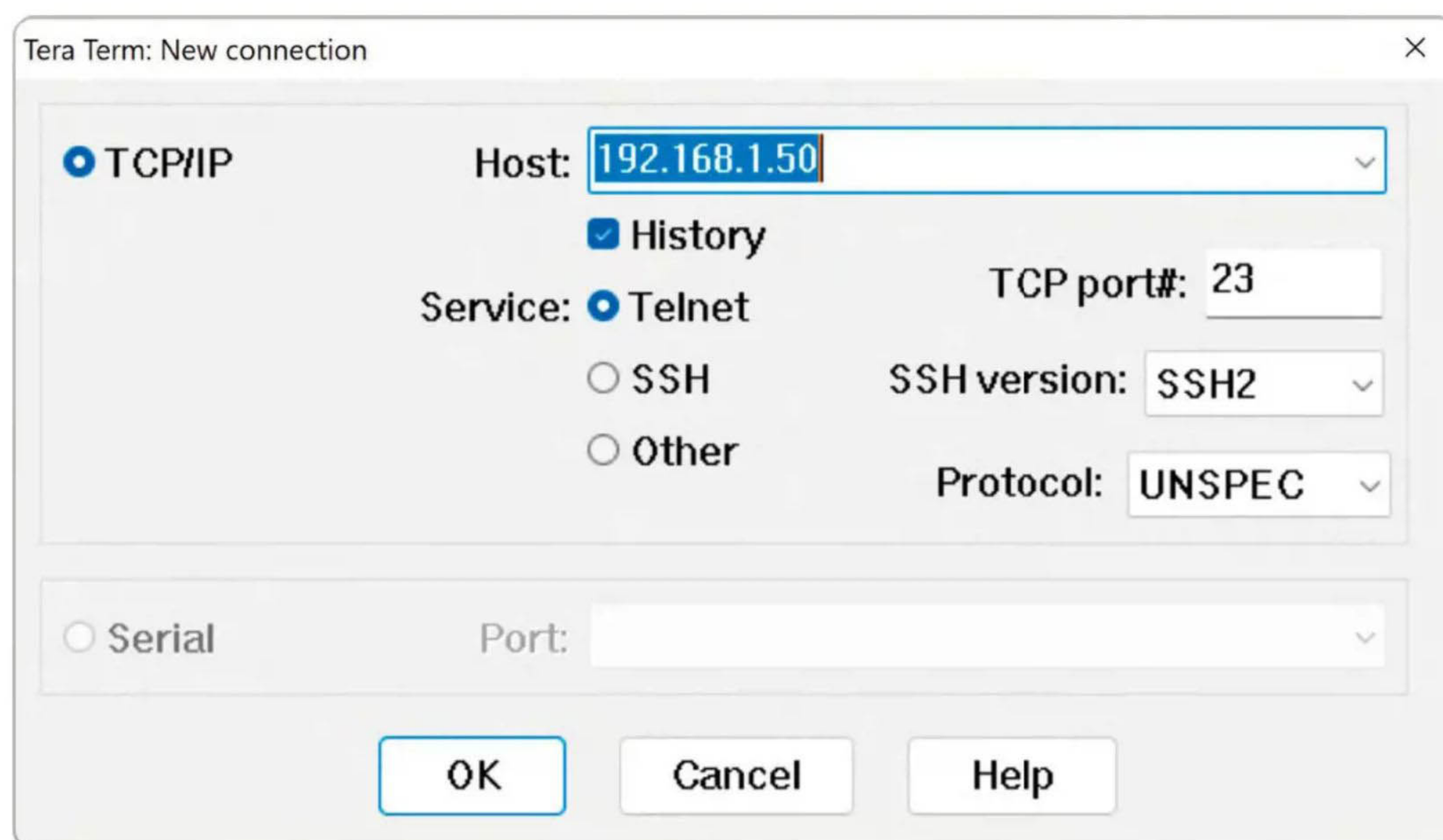
Remote connection via Telnet

The WebMite could be installed in some inaccessible place, so the firmware allows you to use Telnet over your WiFi network to access the MMBasic console. This feature is enabled with the command:

```
OPTION TELNET CONSOLE ON
```

As before, this command will be remembered and automatically applied on every reboot. It will also cause the processor to restart, so you will have to reconnect to regain the command prompt.

The recommended terminal emulator, Tera Term, supports Telnet, so all you need do is select that in the new connection dialog box and enter the WebMite's IP address, as shown in Screen 1. You can do everything you can via a physical USB connection using Telnet, including editing and running programs... all over the WiFi!



Screen 1: You can connect to the WebMite using Telnet over WiFi via Tera Term. This lets you do everything that you can do via a USB cable, including editing and running programs.

You can also use PuTTY in Windows or the telnet command in Linux or macOS (it is no longer part of the macOS by default, but you can install it via Homebrew).

File transfers

Another handy feature is the ability to transfer files to and from the WebMite over WiFi. This is done using TFTP (Trivial File Transfer Protocol) from a Windows, Mac or Linux computer.

In Windows, this is built into the operating system; however, you must enable it first by going to the Control Panel, selecting "Programs and Features", then "Turn Windows features on or off". Finally, scroll down the list and tick TFTP Client.

You can then send a file to the WebMite's internal file system (drive A:) using the following command in a Command or Power Shell window:

```
TFTP -i ipaddress PUT filename
```

This protocol can also be used to retrieve files from the WebMite, eg:

```
TFTP -i ipaddress GET filename
```

Long string support

Another new feature of the WebMite that needs explaining is long strings. Regular string (text) variables in MMBasic can store a maximum of 255 characters. However, most data transferred between an internet client and server is much longer than that. The WebMite has a series of routines called long strings to address this need.

These work with strings of any length, limited only by the available RAM. Using them, you can pull long strings apart, search for specified text, copy parts of the text and so on. They mimic what the standard string functions do in MMBasic, although they

are slower and clunkier (which is why they are only used when required).

To allocate RAM for holding a long string, you need to declare an array of integers with a size (in bytes) that will fit the longest string expected. While declared as integers, the string routines do not store numbers in these; they are just used as memory blocks. For example:

```
DIM INTEGER StrA(512), StrB(512)
```

Each array is 512 integers, and an integer is eight bytes, so each array occupies 4KB of RAM (512B × 8). A character is one byte, so each can store strings of up to 4,096 characters. These arrays are passed to the long string routines using empty brackets. For example, to copy StrB to StrA, you can use the command:

```
LONG STRING COPY StrA(), StrB()
```

Long strings are documented in the user manual. It is worthwhile to familiarise yourself with them as they are invaluable when dealing with the large amount of data sent over the internet.

Web server functions

Assuming that you have connected the WebMite to your WiFi network as described above, the next step in implementing a web server is to tell the firmware to start a TCP server. This is done with another OPTION command as follows (it must be entered at the command prompt):

```
OPTION TCP SERVER PORT 80
```

Port 80 is the standard HTTP port, normally used for serving web pages.

As with the other OPTION commands, this only needs to be entered once and will be remembered. It will also cause the WebMite to restart.

In your program, you tell the server

what to do if an incoming request is received with the command WEB TCP INTERRUPT. This specifies a subroutine that the firmware will call (interrupting the main program) whenever a request is received.

Within your interrupt subroutine, you can retrieve the remote request using the command WEB TCP READ. This command needs a long string buffer for holding the text of the remote request (see above for a description of long strings).

The request from the remote browser will look something like that shown in Screen 2. In this case, the browser is requesting a web page called “page.html”, but it could be the name of an image file or even a single forward slash (/), which is a request for the default page of the website (typically called “index.html”).

The request could also be a notification that the user has clicked on a button or control on the web page and is expecting the WebMite to take some action. In that case, the text between the keywords GET and HTML will indicate the control involved and the user’s action.

Some queries may use the keyword POST instead of GET, but the intention is the same; whatever is between the first keyword and HTML is the request from the remote browser.

Sending a web page

If the request is for a web page, you can send it using WEB TRANSMIT PAGE. This specifies a file formatted in HTML residing in the internal file system of the WebMite (described above) or on an SD card (if connected).

When the firmware transmits the web page, it will scan the page for any embedded BASIC variables surrounded by curly brackets. It will substitute these with the current value of the variable. This facility lets you

insert data your BASIC program has collected into the web page.

For example, if your program had a variable called Humid which had the value of 42 and represented the current humidity, the following text in your HTML file:

The current humidity is {Humid}%

Displays in the client’s browser as:

The current humidity is 42%

You can also send non web pages using the WEB TRANSMIT FILE command, which will send images, audio files and much more (without substituting for variables).

Using these commands, you can create a web server that will serve up pages displaying whatever data you have collected. This server could also respond to remote commands from the user to turn on/off motors, pumps etc, as required. Whatever the WebMite can do locally, it can also do remotely!

A simple web server

This example will display the temperature and humidity in a fictional greenhouse. From the comfort of your living room, you could call up the web page on your phone, tablet or computer and see the current conditions for your plants, even though they may be located at the bottom of your garden.

The whole program is shown in Program 1 and is about as simple as it gets at just 17 lines. The web page is even smaller, at only three lines.

The first line of the program starts the web server and specifies the interrupt subroutine to be used for any incoming requests (“WebInterrupt”).

The next four lines implement a simple loop where a DHT22 sensor (connected to the GP28 pin, as shown in Fig.1) is queried for the current temperature and humidity ten seconds.

The command specifies that the results should be saved in CurrentTemp and CurrentHumid variables.

The WebInterrupt subroutine, starting at line 7, is where the work is done in serving up the web page. Whenever the TCP server receives a request, it will call this subroutine, interrupting whatever the BASIC program was doing at the time.

This subroutine first defines several local integer variables, including an array of integers called b(), used as a long string variable to hold the incoming data.

The web server can handle multiple simultaneous requests, so the program starts a loop stepping through all possible connections. The WEB TCP READ command will read whatever is available on each connection and save any received data in the long string buffer b().

The following two lines look for GET and HTTP keywords in the received request. The next line checks that these keywords are present and in the correct order. In that case, we send the default web page, “index.html”. Note that we don’t care what file the remote browser actually requested; we just send the default page for every request.

This web page is shown in Program 2 and consists of just three lines. The first line is the heading, and the next two define the text on the page. When the page is transmitted, the firmware will substitute the text {CurrentTemp} and {CurrentHumid} with the current values of those variables. “
” in HTML is an instruction to the web browser to insert a new line (line BReak).

Screen 3 shows the result displayed in a browser.

This is a functional program, and if you have a Raspberry Pi Pico W handy, you can copy the files to it and have

```
GET /page.html HTTP/1.1
Host: 192.168.1.50
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7
```

Screen 2: a web request from a remote browser will look like this. The important part is the text between the keywords GET and HTTP. In this case, the browser is requesting a web page called “page.html”, but it could be an image file or some text indicating that the user has clicked a control on the web page.

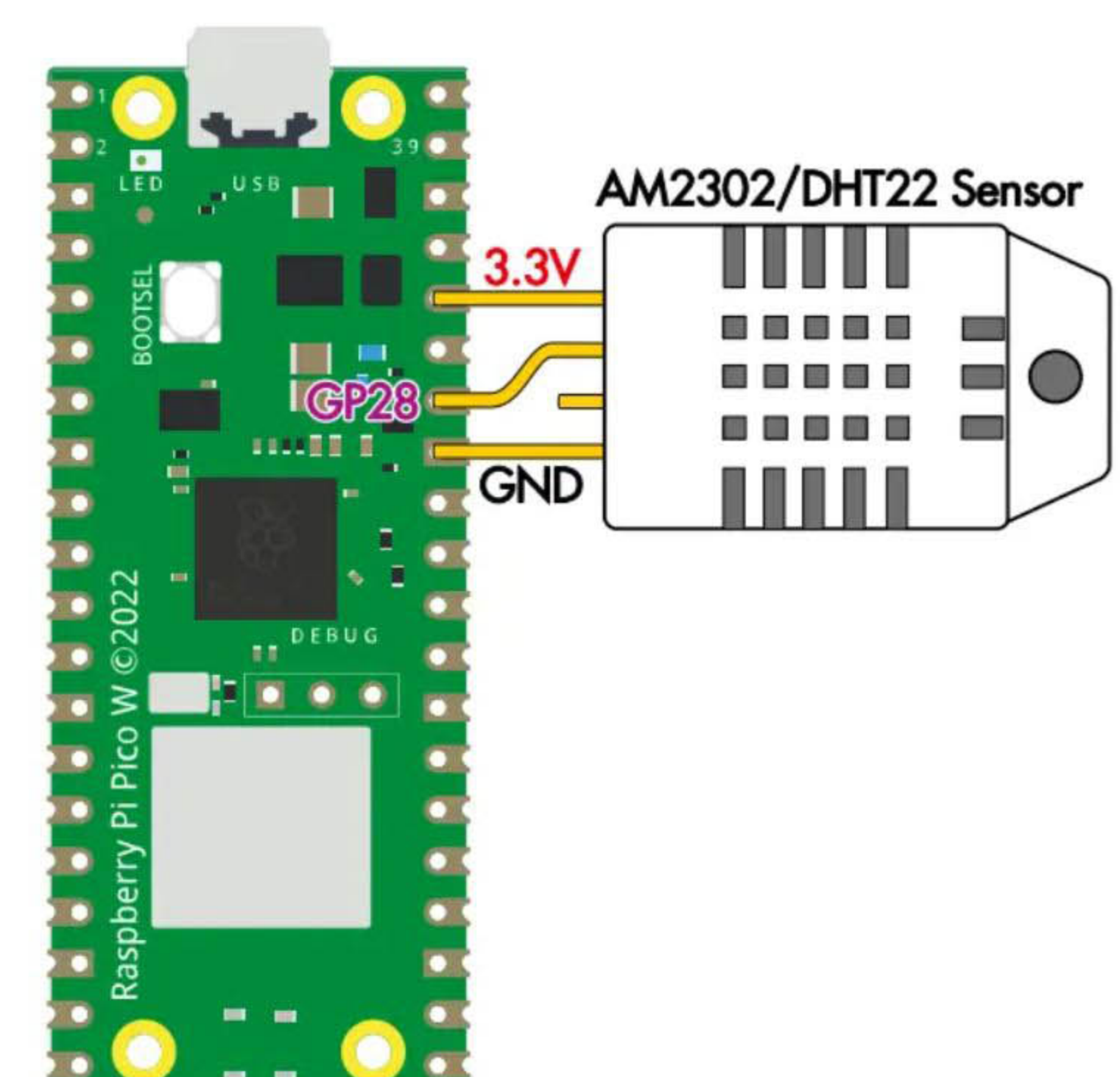


Fig.1: here’s how to connect the DHT22 temperature/humidity sensor for the sample web server program in Program 1.


```

WEB TCP INTERRUPT WebInterrupt
DO
  BITBANG HUMID GP28, CurrentTemp, CurrentHumid
  PAUSE 10000
LOOP

SUB WebInterrupt
  LOCAL INTEGER a, p, t, b(512)
  FOR a = 1 To MM.INFO(MAX CONNECTIONS)
    WEB TCP READ a, b()
    p = LINSTR(b(), "GET")
    t = LINSTR(b(), "HTTP")
    If (p > 0) And (t > p) Then
      WEB TRANSMIT PAGE a, "index.html"
    ENDIF
  NEXT a
END SUB

```

Program 1: this simple web server program displays the temperature and humidity in a fictional greenhouse. It is just 17 lines; the WebInterrupt subroutine starting at line 7 is where the work is done in serving up the web page.

```

<H3>Greenhouse Monitor</H3>
The temperature is {CurrentTemp}&deg;C <BR />
The humidity is {CurrentHumid}%

```

Program 2: this is the web page HTML source for the program shown in Program 1. The first line defines a heading and the next two specify the text in the page. When the page is transmitted, the firmware will substitute the text {CurrentTemp} and {CurrentHumid} with the current values of those variables.

it working immediately. To make it easy, the WebMite firmware download includes both these files and a 'readme' file with detailed instructions. Give it a go and be prepared to be amazed at what the WebMite can do.

Advanced server features

Web infrastructure is a rich environment, so you can add many more features to your web pages. These include images, textured backgrounds, multiple pages and more.

For example, you might want to display a graph of past temperatures and humidity for your hypothetical greenhouse. That can be done by defining a virtual LCD panel in the WebMite. This does not have an attached physical display, but regardless, you can draw your historical data on it using the graphical drawing commands built into MMBasic: line, pixel, text etc.

You can save this image as a BMP file to the internal file system in the WebMite. Then, when a remote browser requests the web page with this embedded image, the browser will also ask for this file, and the user will see an image representing the graph of past temperatures and humidity that your program recorded. Screen 4 illustrates what it could look like.

Even more useful is the ability to define HTML forms in the web page, including embedded controls such as buttons, checkboxes, radio buttons, input text fields and much more. Using these, the user can, via the web page, send commands to the BASIC program running on the WebMite to do things like turn devices off/on, set parameters and so on. Screen 5 provides some examples of these.

A wide range of controls is available but be warned that the HTML code can get complicated. The web page at www.w3schools.com/howto/default.asp lists these controls and their features, and has plenty of examples that you can copy into your web pages.

This ability means that many projects that generally need an LCD screen

with associated buttons and switches (or a touchscreen) can be converted to a web interface with the same functions and more.

An excellent example is the Watering System Controller starting on page 36 of this issue. It uses the WebMite and there are no controls on the physical box. All the controller's functions are configured and controlled exclusively via web pages in a browser.

TCP client

As well as acting as a web server, the WebMite can act as a client and get data from web servers on the internet. Three commands will do this for you: WEB OPEN TCP CLIENT, WEB TCP CLIENT REQUEST and WEB CLOSE TCP CLIENT. For example,

```

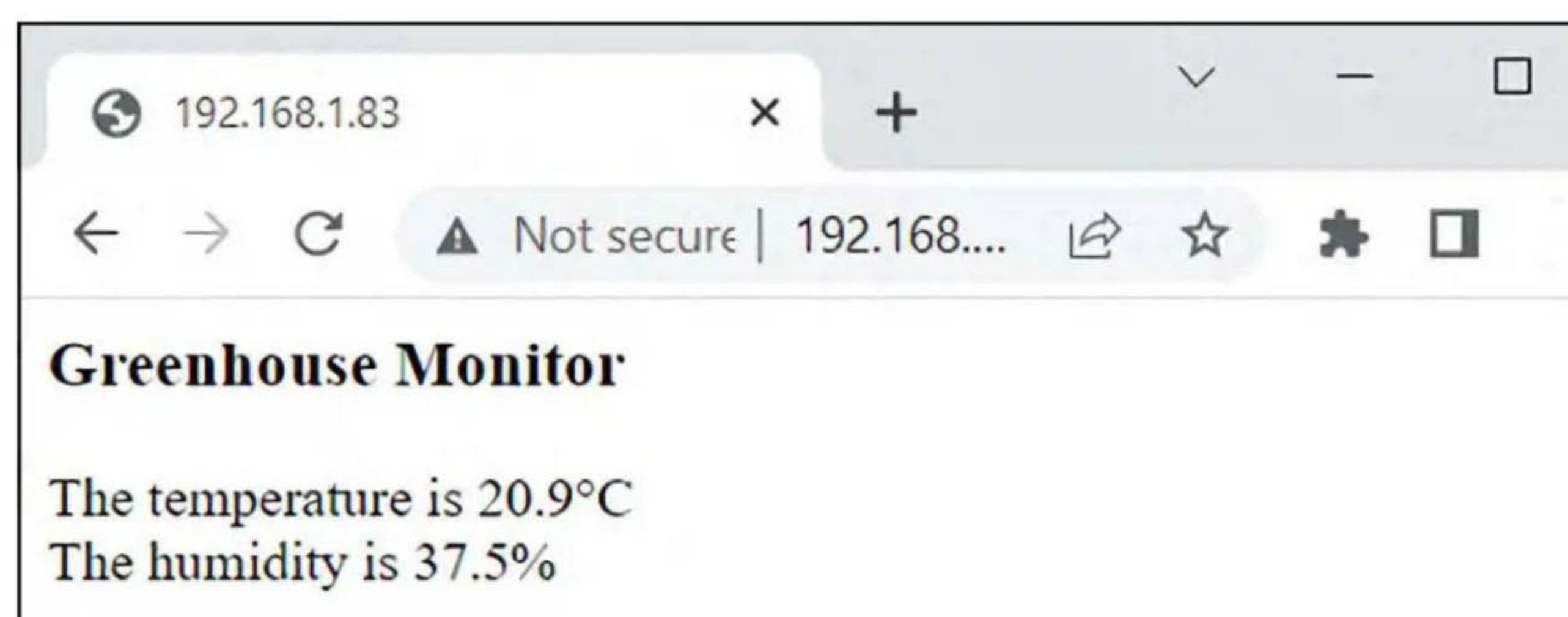
DIM INTEGER b(512)
WEB OPEN TCP CLIENT
  "example.com", 80
WEB TCP CLIENT REQUEST
  "GET / HTTP", b()
WEB CLOSE TCP CLIENT

```

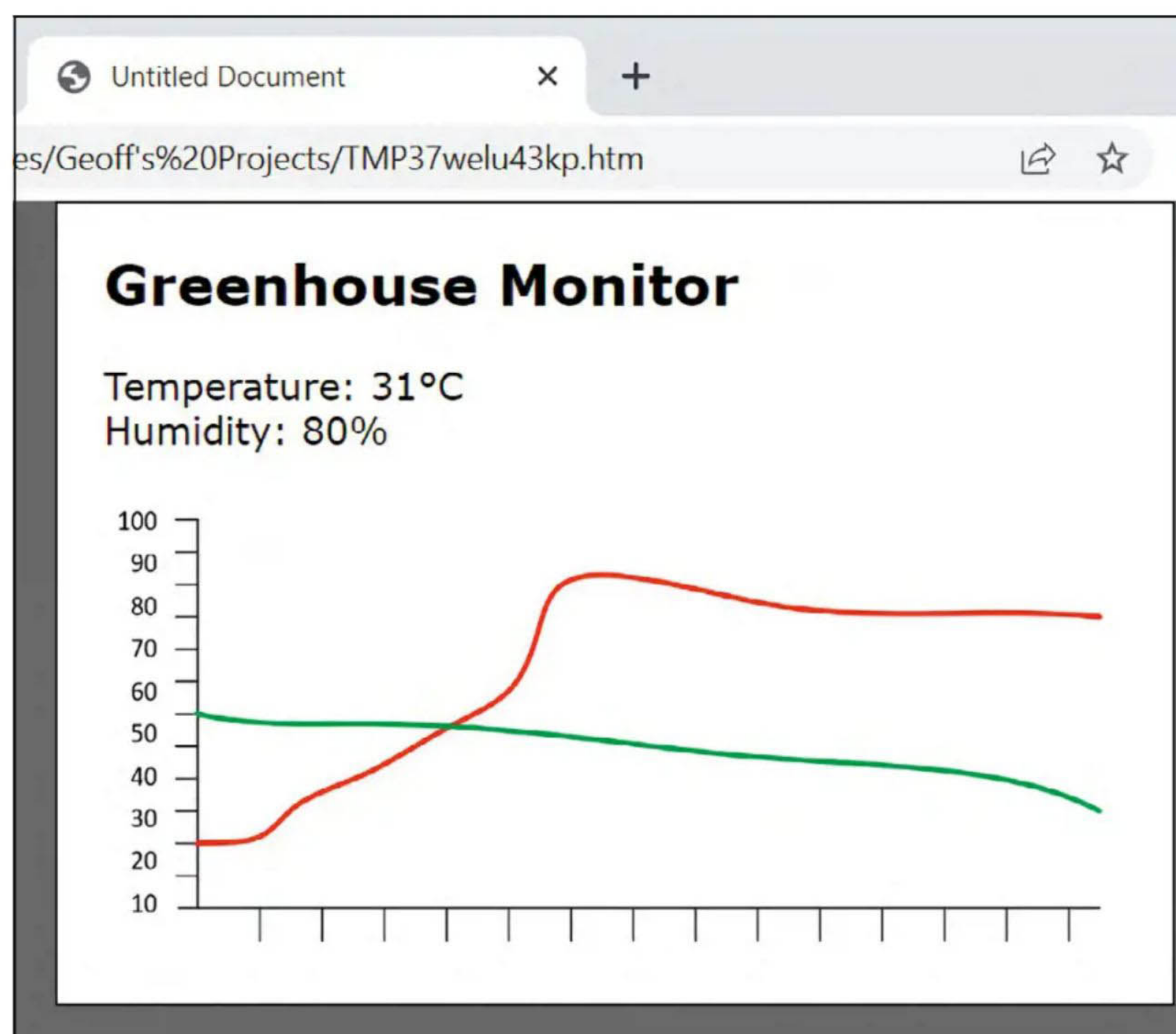
Editor's note: the WEB OPEN and WEB TCP commands should all be on a single line; the commands are shown split here due to limited column width.

The web page would be saved in the long string buffer b() and you could pick it apart to get the data you wanted using the long string routines.

There are many services available

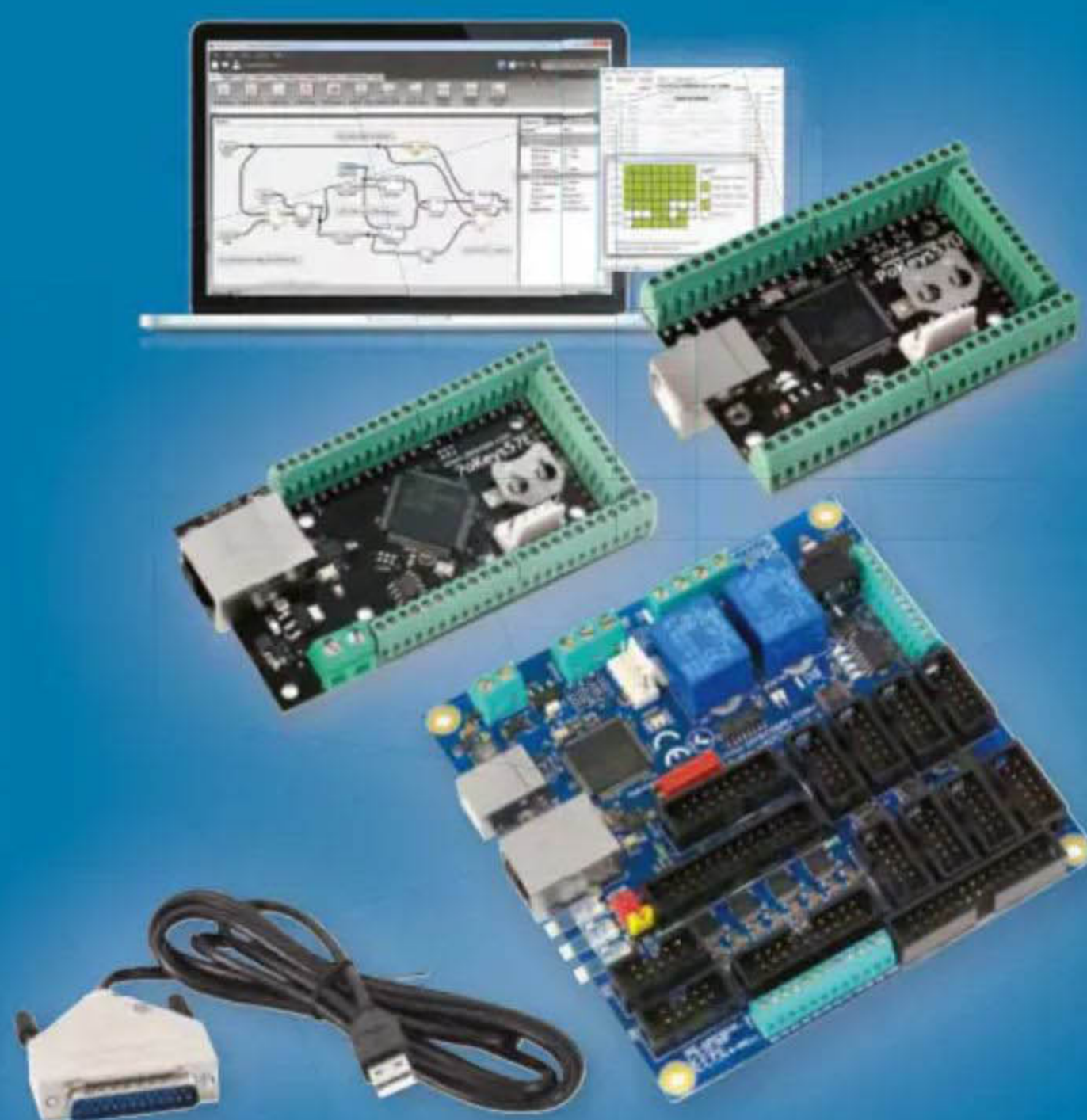


Screen 3: how the simple web server appears to a user on a phone, tablet or computer.



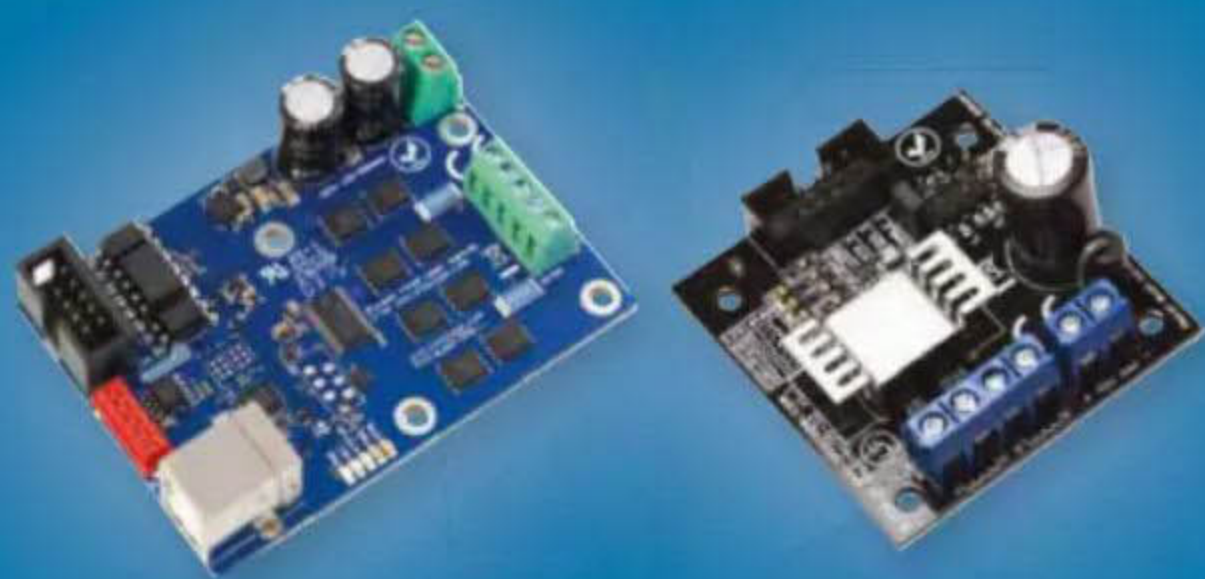
Screen 4: with a little more programming effort, you can extend the simple web server to display a graph of past temperatures and humidity readings for your greenhouse.

PoKeys Connect, Control



- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers Drive



- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+ PoScope Mega50 Measure



- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

on the internet that can be accessed with the TCP client. Two that are documented in the WebMite user manual are getting the weather and sending emails.

You need an account with a weather service to get the weather. The user manual describes how you can connect to Open Weather Map; you can get a wealth of data using their service, such as the current temperature and weather for a city or suburb, plus a forecast for the next day or two. Open Weather Map is free for the basic service and is accurate and comprehensive.

Sending emails is a little more complicated, as most email relay services have protection to stop them from being used to send spam. To send an email, you need to connect to an SMTP relay service that will then send your email to its destination. The example in the user manual uses SendGrid for this task, as they allow a free account to send up to 100 emails a day (plenty for the WebMite).

Sending an email is handy as it allows your WebMite-based gadget to alert you to errors and faults, provide regular status updates etc. For example, your greenhouse monitor could send an email if the temperature went too high or low.

Many internet services now require a secure (encrypted) connection, so the WebMite also implements an experimental version of TLS (Transport Layer Security), an extra protocol layer above the TCP layer that supersedes the SSL protocol. Client-server applications use TLS to communicate across a network in a way designed to

prevent eavesdropping and tampering; many sites insist that this protocol is used.

Network Time Protocol

Getting the current time and date is such a common task that the WebMite has a dedicated command for just this purpose using the Network Time Protocol (NTP) as follows:

WEB NTP timeoffset

With this command, the WebMite will get the date/time from a public time server pool and set the internal clock of the WebMite accordingly. This means that you do not need a real-time clock in your project; you don't need a mechanism to adjust the time or date either.

The parameter "timeoffset" is the local time zone as a floating point number. For example, "WEB NTP 9.5" will get the current time and set the clock in the WebMite to Adelaide time. Note that daylight saving compensation is not included in the NTP service.

Another handy inclusion is a series of commands to post and retrieve data from an MQTT broker. MQTT (Message Queuing Telemetry Transport) is a protocol that enables a client to post data on a server (called an MQTT broker) for later retrieval by another client. It is rather like a bulletin board service for small computers.

An example would be our greenhouse monitor. Say it was battery-powered; it could power up once an hour, measure the temperature/humidity, post the results to an MQTT broker and power down to save the battery. Separately, a client program on a PC could later read these messages, display the results and graph them.

Conclusion

In this summary of the WebMite firmware, we have not mentioned all the features that the WebMite inherited from the PicoMite. That includes the high-performance CPU, fully-featured BASIC programming language, built-in program editor, support for touch-sensitive LCD panels, playing sound and music, external SD cards and an extensive range of communications protocols like serial and I²C.

The Raspberry Pi Pico W costs little (~£5) and is readily available, so why not have a go? Load the free WebMite firmware and discover how easy it is to connect to the internet or turn this cheap device into a web server.

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au

☒ Radio 1
 ☐ Radio 2

BUTTON 1

BUTTON 2

☐ Checkbox 1
 ☒ Checkbox 2

Enter Text

Slider:

Countdown: 4h 21m 19s

Screen 5: you can include embedded controls such as buttons, checkboxes, radio buttons, input text fields and much more on a web page using HTML forms. The user can use these to send commands to the BASIC program running on the WebMite to turn devices off or on, set parameters and so on.

ETI BUNDLE (1) Teach-In 3, 4 and 5 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 3 – CD-ROM

Mike & Richard Tooley

The three sections of the *Teach-In 3* CD-ROM cover a huge range of subjects that will interest everyone involved in electronics – from newcomers to the hobby and students to experienced constructors and professionals.

The first section (80 pages) is dedicated to *Circuit Surgery*, EPE/PE's regular clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

The second section – *Practically Speaking* – covers hands-on aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and identifying components is covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by readers.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge and TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The *Teach-In 1* series covers everything from electric current through to microprocessors and microcontrollers, and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

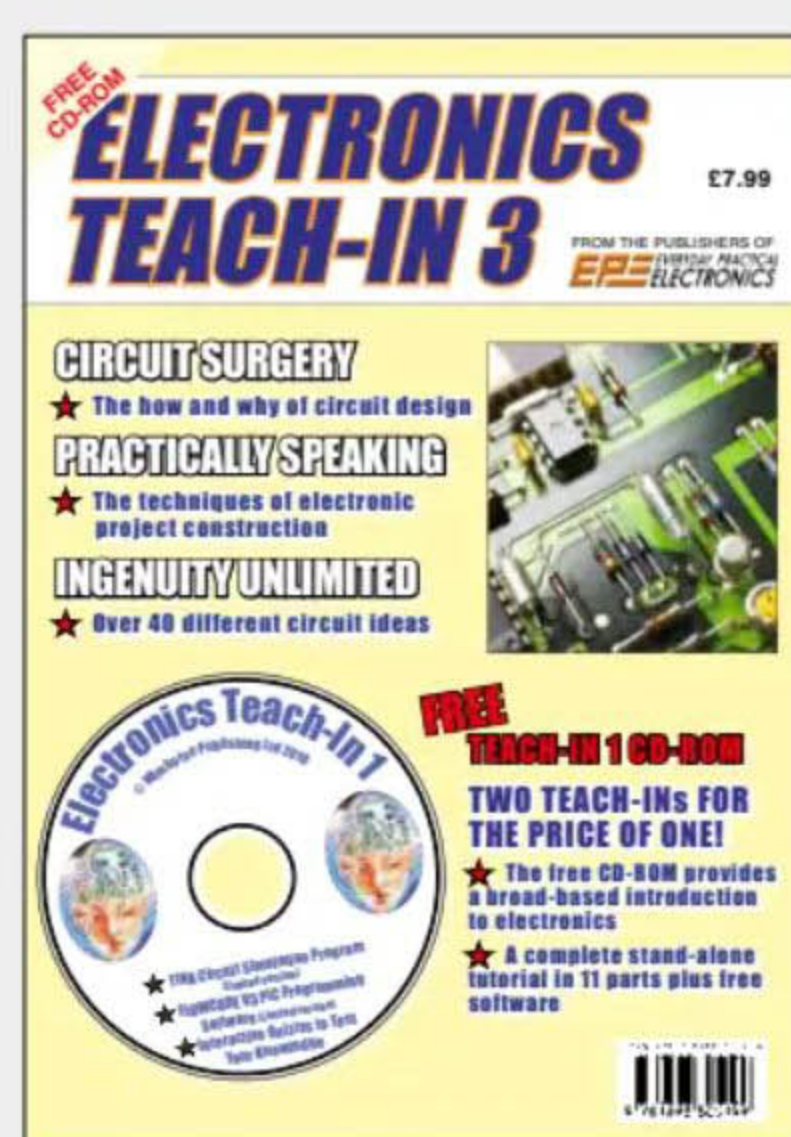
ELECTRONICS TEACH-IN 4 – CD-ROM A BROAD-BASED INTRODUCTION TO ELECTRONICS

Mike & Richard Tooley

The *Teach-In 4* CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full *Modern Electronics Manual*, worth £29.95. The **Manual** contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to anyone interested in learning about electronics – hobbyists, students or professionals.



ELECTRONICS TEACH-IN 5 – CD-ROM JUMP START

Mike & Richard Tooley

15 design and build circuit projects for newcomers or those following courses in school and colleges.

The projects are: ■ Moisture Detector ■ Quiz Machine ■ Battery Voltage Checker ■ Solar-Powered Charger ■ Versatile Theft Alarm ■ Spooky Circuits ■ Frost Alarm ■ Mini Christmas Lights ■ iPod Speaker ■ Logic Probe ■ DC Motor Controller ■ Egg Timer ■ Signal Injector Probe ■ Simple Radio Receiver ■ Temperature Alarm.

PLUS

PIC'n' Mix – starting out with the popular range of PIC microcontrollers and *Practically Speaking* – tips and techniques for project construction.

The CD-ROM also contains:

- Complete *Teach-In 2* book, a practical introduction to PIC microprocessors
- MikroElektronika, Microchip and L-Tek PoScope software.



Three Teach-ins for the great price of
£18.95

PLUS you also get the contents of the
free CD-ROM from each issue...

... so that's another TWO Teach-Ins and
The Full Modern Electronics Manual!

What a Bargain!!

ETI BUNDLE (2) Teach-In 6, 7 and 8 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 6 – CD-ROM A COMPREHENSIVE GUIDE TO RASPBERRY PI

Mike & Richard Tooley

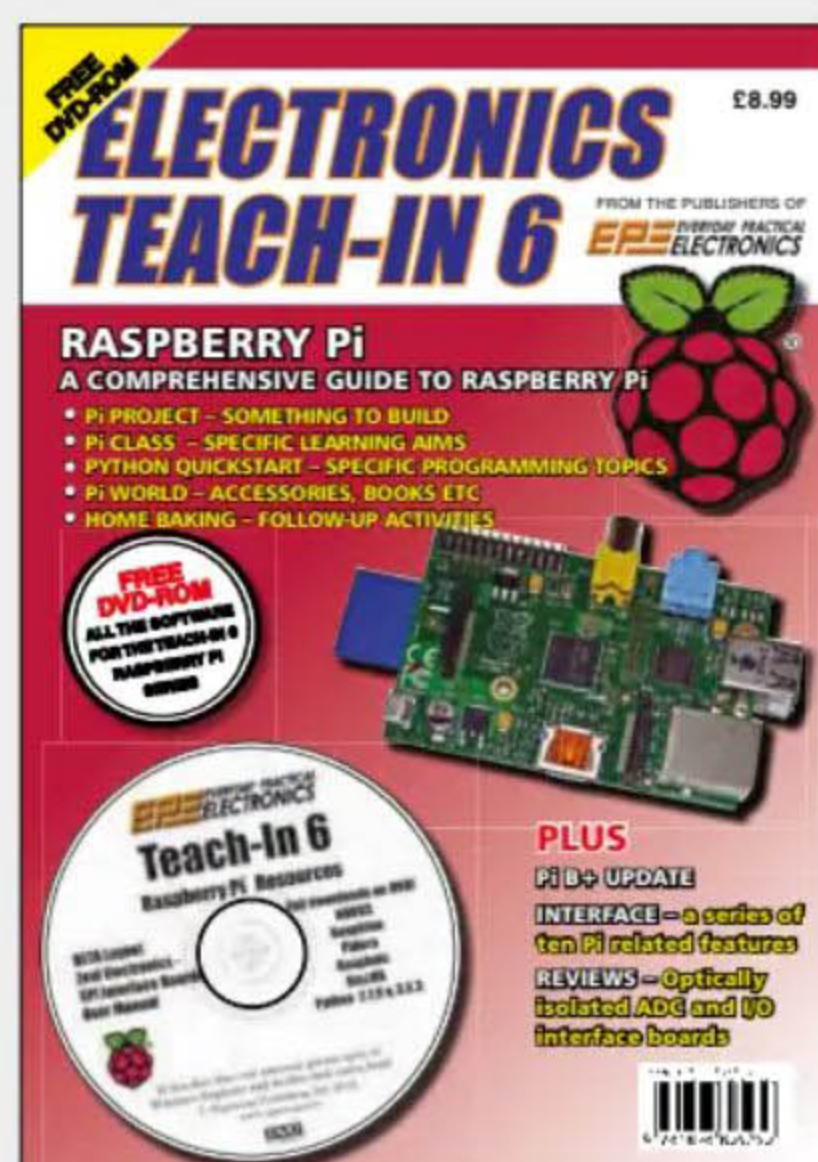
Teach-In 6 contains an exciting series of articles that provides a complete introduction to the Raspberry Pi, the low cost computer that has taken the education and computing world by storm.

This latest book in our *Teach-In* series will appeal to electronics enthusiasts and computer buffs who want to get to grips with the Raspberry Pi.

Teach-In 6 is for anyone searching for ideas to use their Pi, or who has an idea for a project but doesn't know how to turn it into reality. This book will prove invaluable for anyone fascinated by the revolutionary Pi. It covers:

- Pi programming
- Pi hardware
- Pi communications
- Pi Projects
- Pi Class
- Python Quickstart
- Pi World
- ...and much more!

The *Teach-In 6* CD-ROM also contains all the necessary software for the series, so that readers and circuit designers can get started quickly and easily with the projects and ideas covered.



ELECTRONICS TEACH-IN 7 – CD-ROM DISCRETE LINEAR CIRCUIT DESIGN

Mike & Richard Tooley

Teach-In 7 is a complete introduction to the design of analogue electronic circuits. It is ideal for everyone interested in electronics as a hobby and for those studying technology at schools and colleges. The CD-ROM also contains all the circuit software for the course, plus demo CAD software for use with the *Teach-In* series.

- Discrete Linear Circuit Design
- Understand linear circuit design
- Learn with 'TINA' – modern CAD software
- Design simple, but elegant circuits
- Five projects to build:
 - i) Pre-amp
 - ii) Headphone Amp
 - iii) Tone Control
 - iv) VU-meter
 - v) High Performance Audio Power Amp.

PLUS

Audio Out – an analogue expert's take on specialist circuits

Practically Speaking – the techniques of project building.



ELECTRONICS TEACH-IN 8 – CD-ROM INTRODUCING THE ARDUINO

Mike & Richard Tooley

Hardware: learn about components and circuits

Programming: powerful integrated development system

Microcontrollers: understand control operations

Communications: connect to PCs and other Arduinos.

Teach-In 8 is an exciting series designed for electronics enthusiasts who want to get to grips with the inexpensive, popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. It will provide a one-stop source of ideas and practical information.

The Arduino offers a truly effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle through wireless or the Internet.

Teach-In 8 is based around a series of practical projects with plenty of information for customisation.

This book also includes *PIC n' Mix*: 'PICs and the PICkit 3 – A Beginners guide' by Mike O'Keefe and *Circuit Surgery* by Ian Bell – 'State Machines part 1 and 2'.

The CD-ROM includes the files for:

- *Teach-In 8*
- Microchip MPLAB IDE XC8 8-bit compiler
- PICkit 3 User Guide
- Lab-Nation Smartscope software.



ORDER YOUR BUNDLE TODAY!

JUST CALL 01202 880299 – OR VISIT www.electronpublishing.com

WEBMITE-BASED WATERING SYSTEM CONTROLLER

BY GEOFF GRAHAM

This advanced Watering System Controller does it all. It can change the watering schedule depending on the seasons, check the weather forecast before watering and even alert you to a burst pipe or blocked sprinkler. Best of all, it is very easy to set up and use.

Wateringsystemcontrollers, often known as reticulation or irrigation controllers, are notoriously difficult to program. They generally have a small LCD screen and an array of tiny buttons or switches to configure the watering schedule.

Our Watering System Controller has no LCD or buttons; it is entirely set up and controlled via a web browser on your smartphone, tablet or computer. The web pages are easy to understand and provide everything you need to set it up.

The key component is the WebMite, a Raspberry Pi Pico W microcontroller running the MMBasic programming language, starting on page 28 in this issue. Here it delivers the web pages, gets the time, date and weather from the internet and follows the watering schedule you have configured.

The only other significant components in this design are a small power supply and the solid-state relays (SSRs) that drive the solenoid watering valves.

The program running on the WebMite is written in BASIC, so you can

read it and edit it if you have any special requirements.

This project was partly developed to demonstrate what you can do with the WebMite. Even if you do not want to build a Watering System Controller, this project can be helpful by providing the code and techniques you can use in other projects.

The Watering System Controller program will even run on a bare Raspberry Pi Pico W. That means you can explore the program and try it yourself without building anything; you just need the Pico W module.

Watering system layout

Fig.1 shows a typical reticulation layout. The water supply is connected via a master valve to a series of valves that control individual sprinklers (or sets of sprinklers). The Controller opens the master valve, then steps through each sprinkler valve in turn, opening them for the specified time.

The master valve is important as it provides a backup if one of the sprinkler valves is stuck open (it happens). The master valve will still cut off the water supply, limiting the

damage caused by the permanently-open valve.

Some systems have a pump as the water supply; in that case, the Watering System Controller would switch that on and off instead of a master valve.

Either way, the Controller will switch it on half a second before the first sprinkler valve is opened and switch it off half a second after the last has been closed. That is to ensure that the water pipes leading to the sprinkler valves are kept full of water, avoiding the situation where a sprinkler valve closes after the master valve, letting water out of the pipes.

Otherwise, when the master valve next opens, the water rushing through the empty pipes could act as a hammer and damage the joints in the pipe and the valve.

The flow sensor is an optional but worthwhile addition. It is usually fitted at the outlet of the pump or master valve and tells the Controller the amount of water flowing through the system. The Controller records this and, over time, builds up an average flow rate for each valve.

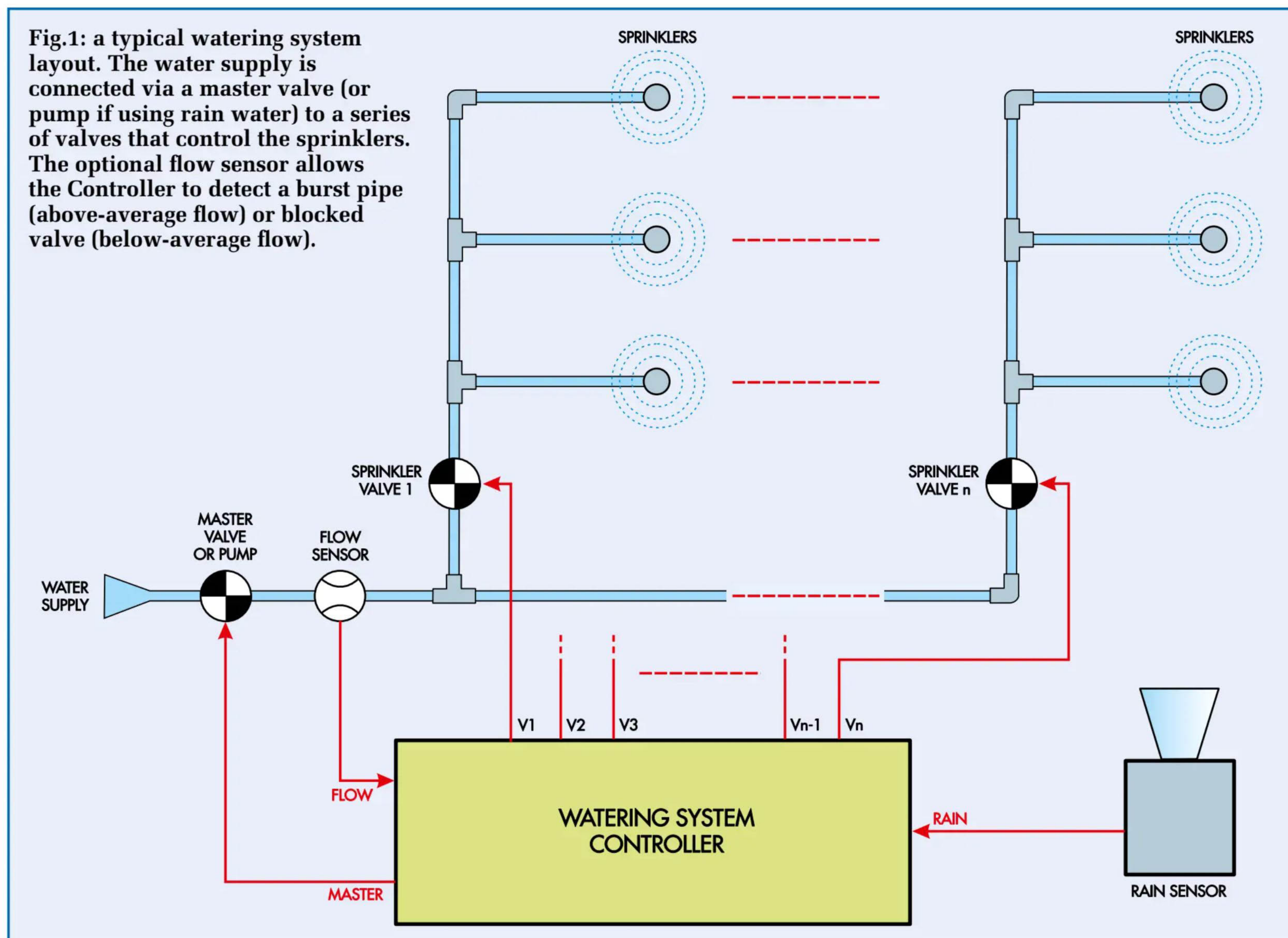
The Controller can then easily detect an abnormal flow rate when it opens a valve, either over or under the average. This is invaluable as an excessive flow indicates that a pipe has burst and it might be digging a hole in your garden. A lower-than-normal flow rate means a sprinkler or valve is blocked; if left unfixed, that could cause your plants to die.

When an excessively high flow is found, the Controller immediately shuts off that valve to stop any damage. In the case of underflow, it will continue with the watering time in the hope that some water is getting



The finished controller in a weatherproof enclosure mounted on an exterior wall. We decided to mount the status LED on the lid along with a waterproof power switch.

Fig.1: a typical watering system layout. The water supply is connected via a master valve (or pump if using rain water) to a series of valves that control the sprinklers. The optional flow sensor allows the Controller to detect a burst pipe (above-average flow) or blocked valve (below-average flow).



through, but it will also inform you of the problem.

The rain sensor is also optional. The Watering System Controller can check the weather forecast for you and skip watering if rain is predicted, but connecting a rain sensor provides a backup specific to your garden. The Controller will check the sensor before it starts a watering run, and if it indicates that it is currently raining, the schedule will be skipped for that day.

Controller capabilities

To set up the Controller, you use a web browser to access its web page, shown in Screen 1. It lists all the possible watering schedules, with a button to configure each. It also has a RUN NOW button that can be used to start the watering schedule at any time.

These schedules are set to start at a certain date in the year and terminate at another. For example, you can set up a schedule for the summer months tailored to the demands of hot weather, another for autumn when less watering is required, another for winter and so on. You can set up the watering requirements for the whole year and, unless you change your mind, it will be repeated year after year.

Typical Watering System Controllers require you to change the watering characteristics at the start of each season, and that can quickly become tedious, something that our design cleverly eliminates.

That page includes buttons to configure each schedule. After clicking one, you will be presented with a web page similar to that shown in Screen 2, which provides all the schedule details.

The first field on the configuration page allows you to enter a meaningful title to be displayed on the main page. You can also enable or disable the schedule with a checkbox. The next four fields let you set the start and stop dates in the year for the schedule. Schedules can overlap and, if you want the same watering scheme for the whole year, you can set the start to 1/1 and the end to 31/12.

Features & Specifications

- » Configured using a web interface from a smartphone, tablet or computer
- » Controls a master valve plus up to eight sprinkler valves
- » Up to eight independent watering schedules
- » Schedules can be customised for different requirements during summer etc
- » The active schedule can be skipped if rain is forecast or detected
- » Sprinkler times can be increased on hot days
- » Flow sensor support for detecting burst pipes, blocked valves or sprinklers
- » A rain sensor can be connected to avoid watering during rain
- » Schedules can run on certain days of the week or at intervals in days
- » Schedules can start at a fixed time or a period before/after sunrise/sunset
- » Time and date are set from the internet with automatic daylight saving compensation
- » No battery-backed clock is required
- » Schedules continue if the WiFi or internet is down
- » Schedules automatically restart after a power failure
- » Powered by 24V AC at 1A

The following section allows you to set the days of the week for watering or specify a watering interval in days. This is a case of one or the other but not both.

Following this, you set the time for the watering to start. Screen 2 shows this set to 6 hours 0 minutes after midnight (ie, 6am). However, using the dropdown list, you can also specify a number of hours and minutes before sunrise, after sunrise, before sunset or after sunset. That gives you a lot of flexibility in setting the start time.

In the next section, you can configure each solenoid valve’s watering time in minutes. The Controller will step through each valve in the sequence, opening it for the specified time. It can drive up to eight

valves (plus the master valve), but you do not need to install that many if you need fewer. The program will skip any valves with a zero or blank watering time.

If you have entered some watering times in these fields, the RUN NOW button will appear on the main page, even if the schedule is disabled and the other fields have not been filled in. That means you can create a watering sequence that can only be started on demand from the web page.

The final section allows you to change the watering schedule for that day depending on the weather forecast. You can skip watering entirely if rain is forecast, and you can increase the watering times if the forecast for the next 24 hours predicts

a maximum temperature over a certain threshold. The former can avoid wasting water, while the latter can prevent plant death on unusually hot days.

At the bottom of the page is a button to save the changes that you have made. The BASIC program saves the configuration and settings to a file called “settings.dat” in the internal file system of the WebMite. On power-up or a reboot, the program reads that file so that all the details are in memory.

General settings

The GENERAL SETTINGS button at the bottom of the main page (in Screen 1) takes you to the web page shown in Screen 3. On this page, you can tell the program your location, connected sensors and details for sending emails.

The location data is used for many features of the Controller, including its time zone, daylight saving compensation, the times for sunrise and sunset and the weather forecast. You need to enter the name of your city and the country code, which is AU for Australia and NZ for New Zealand (a full list of the Alpha-2 codes can be found at [https://w.wiki/Gb\\$](https://w.wiki/Gb$)). After that, you can click on the TEST button, and you should see a response similar to that shown in Screen 4.

For large cities, you can also enter a suburb (eg, “North Sydney”); if that is found, it will provide you with a more specific weather forecast.

The database has over 200,000 cities and towns, so you should be able to find your location. If you cannot, try for the nearest larger city or town within the same time zone. The weather predictions might still be accurate enough, depending on how close it is to you.

These functions use data from Open Weather Map (<http://openweathermap.com>). It uses that service to look up the latitude and longitude when validating your location. The program can then use that information to query Open Weather Map for your time zone (including DST) and the sunrise and sunset times.

If a schedule depends on the weather forecast, the program will query Open Weather Map for the 24-hour forecast before running the watering schedule. With that data, the program can then determine if the schedule should be skipped or needs to be modified.

If you do not enter a location, the Controller will still operate but will use the AEST (GMT + 10 hours) time

Screen 1: the main web page you see when you connect to the Controller. It lists all the watering schedules and their start and stop dates. The CONFIGURE button lets you change the associated schedule, while the RUN NOW button immediately starts that program.

SILICON
CHIP

Watering System
Controller

Summer (morning)	13/11 to 31/3	CONFIGURE	RUN NOW
Summer (evening)	13/11 to 31/3	CONFIGURE	RUN NOW
Autumn	1/4 to 31/5	CONFIGURE	RUN NOW
Winter (maintenance)	1/6 to 31/8	CONFIGURE	RUN NOW
Spring	1/9 to 12/11	CONFIGURE	RUN NOW
unused	Disabled	CONFIGURE	
unused	Disabled	CONFIGURE	
Test (2 Min Each)	Disabled	CONFIGURE	RUN NOW

GENERAL SETTINGS

MMBasic Ver 5.0707 retic.bas Ver 1.0
<http://geoffg.net/retic.html>
<http://siliconchip.com.au>

Screen 2: this allows you to modify a schedule, including the start and stop dates, the time to start watering, the watering time for each valve and the actions to take based on the weather forecast.

SILICON
CHIP

Configure Schedule 1

Schedule: Title (max 20 characters)
☒ Enable this schedule

Active Period (inclusive):
Start date: Day Month
End date: Day Month

Watering Days:
☐ Su ☒ Mo ☐ Tu ☐ We ☐ Th ☒ Fr ☐ Sa
-or-
run every days

Start Watering: hours minutes

Run Time (minutes):
Valve 1 Valve 2 Valve 3 Valve 4
Valve 5 Valve 6 Valve 7 Valve 8

Weather Forecast:
☒ Skip schedule if rain is forecast in the next 24 hours
Add % to run time if forecast max is over °C

SAVE CHANGES

RETURN TO HOME PAGE

zone or whatever is set in the BASIC program. You can still set a watering schedule and start times, but compensation for daylight saving will not be included, and you will not be able to set times based on sunset/sunrise or modify the schedule based on the weather.

Sensors

There is a section below the location data where you can configure the flow and/or rain sensors (if fitted).

The flow sensor will alert you if the water flow for a particular valve is significantly over or under its average flow rate. Either case will cause the status LED to flash and add a warning message on the main web page of the Controller. However, these can be missed, so an email alert (see below) should also be configured to ensure you are notified of the fault.

If the fault is not corrected, the Controller will, over time, add this abnormal flow into its average flow rate for the valve and eventually stop treating it as a failure. So, if you get an email indicating a fault, make sure that you attend to it. When you fix the fault, click the button to reset the average so that the BASIC program knows to build a new average for fault detection (it will do that for all valves).

You can connect a rain sensor and, if configured, the Controller will not run a watering schedule if it is currently raining. That is in addition to checking the weather forecast. Most rain sensors have normally-closed contacts that open in the case of rain, which is what the Controller is designed for. You just need to connect it to the screw terminal on the Controller and tick the box to enable this function.

Sending emails

This is an invaluable feature as you would not normally check the Controller’s status LED or web page daily. Because sprinklers are usually run before sunrise, critical faults can remain unnoticed for months, by which time they could have done a lot of damage to your garden.

To send emails, you need to open a free account with SendGrid (<http://sendgrid.com>). Opening the account is a little tedious because they need to verify your identity to prevent spammers from abusing their service. However, with the account created, you can get a free API key (a 69-character string) that you can enter in the API key field.

When you get the key from SendGrid, you must also provide them with a matching “from email address”,



General Settings

Location:

The location is required to determine the time zone, daylight saving, sun rise/set and the weather. If it is not set UTC time will be used.

City Name: Country Code:

The country code is generally the first two letters of the country (called the ALPHA-2 code), [click here for the full list](#). Use the TEST button to test the settings. If you cannot find your city try a nearby larger city and if that fails chose a city in the same timezone with the same daylight saving rules.

TEST

Sensors:

Detect Abnormal Water Flow: ☐ Enable Historical average

Installed Rain Sensor: ☐ Enabled (contacts open on rain)

Detecting abnormal water flow requires that a flow meter be installed at the pump or master valve outlet with its output signal connected to the flow input terminal.

Email Settings:

This setting is optional but is required if email notifications are selected below.

API Key:

From Email Address:

To Email Address:

Notifications:

☒ Send email if unable to get the current time/date

☒ Send email if unable to get the weather forecast

☐ Send email if abnormal water flow is detected

☒ Send a weekly email with the current status

Open a free account with Twilio SendGrid (<https://sendgrid.com>) then create an API key (a 69 character string starting with SG) and paste it into the API Key field above. Note that the From Address must be the same as that registered with the SendGrid API key. The To address can be the same or different.

Screen 3: this screen lets you change the settings for the Controller as a whole. You can tell it your location, the connected sensors and how to send emails. The location is particularly important as it is used to determine the time zone, daylight saving compensation, times for sunrise and sunset and the weather forecast.



General Settings

Location:

The location is required to determine the time zone, daylight saving, sun rise/set and the weather. If it is not set UTC time will be used.

City Name: Country Code:

Success.
Found Perth, Western Australia, AU

The country code is generally the first two letters of the country (called the ALPHA-2 code), [click here for the full list](#). Use the TEST button to test the settings. If you cannot find your city try a nearby larger city and if that fails chose a city in the same timezone with the same daylight saving rules.

TEST

Screen 4: you can test the location you entered and if it is found, you will see a message like this. When you test the email function, you will see a similar message confirming that it worked OK.

which should be entered in the next field (From Email Address). Finally, you need to provide an email address to receive emails. This can be the same as the From Address or different.

To test your settings, click on the TEST button. After a few seconds, you should see a confirmation message telling you that a test email has been successfully sent. You can then check your email inbox to confirm you have received it.

Circuit details

The circuit diagram for the Watering System Controller, Fig.2, is

dominated by the Raspberry Pi Pico W (ie, the WebMite). The only other significant components are the power supply and the solenoid drivers.

All valves use a single common return connection. The Controller is designed for the typical solenoid valves used in domestic reticulation systems that are controlled by 24V AC. These solenoids usually draw a surge current of 350mA when energised, then drop to a holding current of about 220mA.

An IXYS CPC1965 solid-state relay drives each valve. These switch on the zero crossing of the AC waveform,

so there are no problems with inductive kickback from the coils in the solenoid valves. These are controlled by an inbuilt LED that provides isolation between the input and output.

The drive current for the LED is about 5mA (limited by the 470Ω resistor), which is well within the drive capability of the microcontroller's digital outputs.

Only two solenoid valves can be energised at once: the master and the currently open sprinkler valve. These are protected by separate PPTC (polymeric positive temperature coefficient) 'fuses', which increase in resistance if there is an excessive current through them, limiting the maximum current. When the fault is removed, they revert to regular operation. They protect against short circuits in a solenoid or the solenoid wiring.

The power supply is a switching buck (step-down) regulator providing 5V DC to the WebMite and the flow sensor (if fitted). This comprises switching regulator REG1, inductor L1, diode D1, a feedback voltage

divider that sets the output to 5V and a couple of bypass/filter capacitors. The power requirement of the Controller is modest at 60-100mA.

However, the relatively high input voltage of around 34V DC from the rectified 24V AC would result in 3-4W of heat being generated by a linear regulator. In a sealed enclosure, that could lessen the life of the electrolytic capacitors. With the switching power supply, dissipation is less than 1W.

The life of the electrolytic capacitors is a major consideration, as the Watering System Controller should ideally last for 10-20 years or more. For this reason, we have specified high-voltage, high-temperature capacitors with higher capacitances than strictly necessary. Those factors together should extend the life of the capacitors considerably.

5V power to the WebMite is supplied via schottky diode D2. This is to isolate the Controller's power supply from the USB 5V provided by your computer if you have plugged that

into the WebMite, letting you use the USB port to debug and test the software on the WebMite even while the Watering System is powered.

The flow and rain sensor inputs are pulled up to 3.3V (from the Pico) by 3.3kΩ resistors and clamped to stay within the supply rails by pairs of schottky diodes. The diodes are for protection from miswiring, nearby lightning strikes etc. In both cases, the input is pulled to ground by the sensor, which is detected and processed by the WebMite's BASIC program.

There are two LEDs and two tactile switches mounted on the PCB. The red LED indicates the controller status; when it is solidly lit, the Controller is operating without fault. If it is flashing or off, that indicates a fault like an abnormal flow detected by the flow sensor, an inability to connect to the internet etc.

The green LED illuminates when the Controller is running a watering cycle; the abort button below it can terminate this cycle. The reset switch will force the WebMite to reboot, which is useful if you are upgrading the firmware.

Circuit board design

The fully-populated Watering System Controller PCB is shown in Photo 1. This is intended to be mounted in a RITEC RP1285BF 186 × 146 × 75mm waterproof sealed enclosure. Altronics stocks this (Cat H0310F) and there are others of a similar size, some with a clear lid.

Note the screw terminals along the bottom edge of the PCB. The first on the left is the 24V AC power input. While a capacity of 1A is specified, a source capable of providing 750mA or more should work OK.

Further along the bottom are the outputs for driving the eight sprinkler valves plus one master valve.

Also on the bottom edge are the inputs for the optional flow and rain sensors. The flow sensor should be a Hall effect type that can be powered by 5V DC. The parts list gives a typical example, although there are many other suppliers. The rain sensor does not need power and should be a type with normally-closed contacts; again, the parts list gives a typical example.

The board has been designed so there is plenty of space between the screw terminals and the bottom of the case, allowing you to route the wires easily.

On the top of the PCB, there is space for a small toroidal transformer that can be installed by a qualified

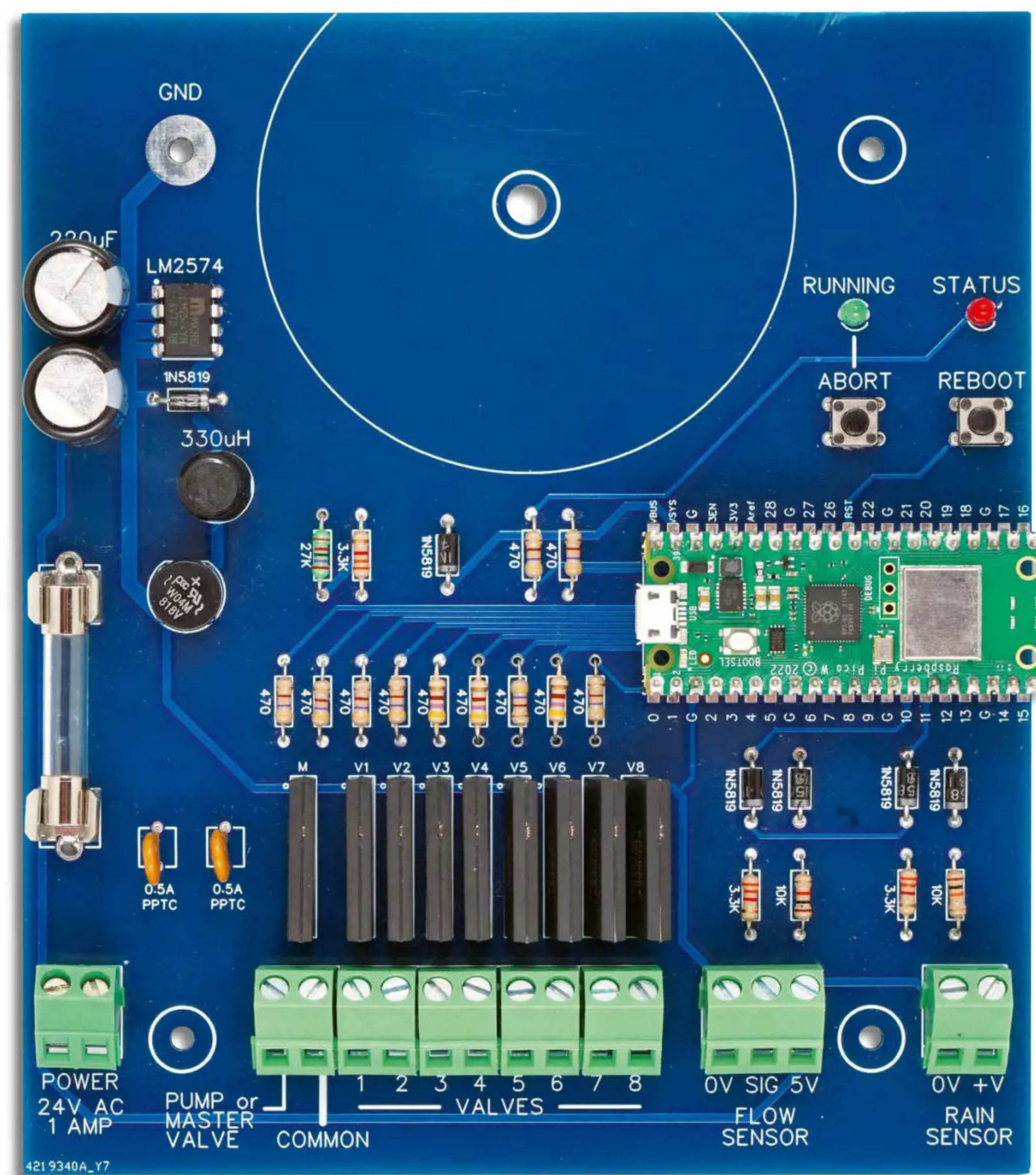
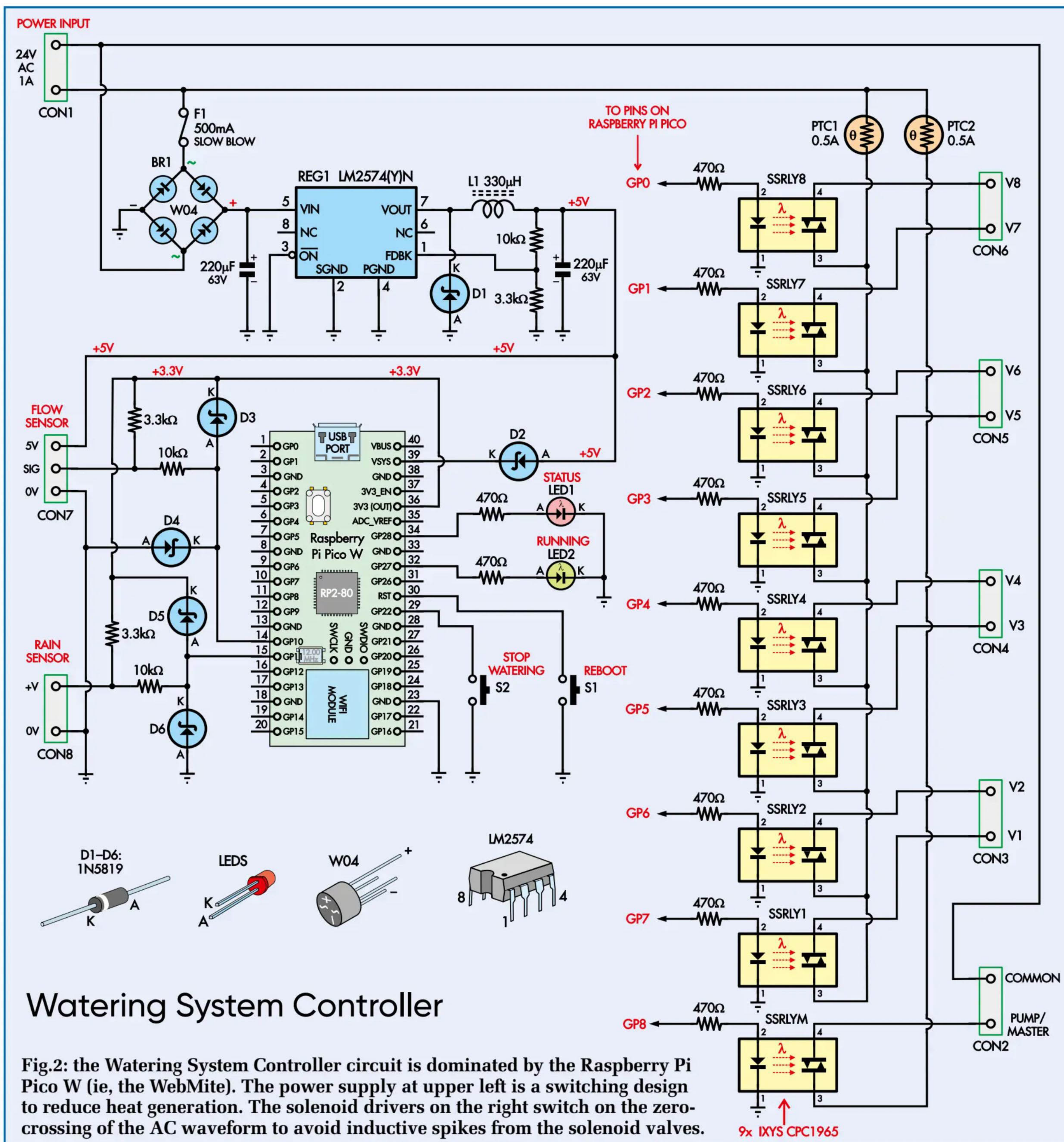


Photo 1: The fully populated Watering System Controller PCB (shown smaller than actual size). Along the bottom edge are the screw terminals for the power input, the master valve plus eight sprinkler valves and the inputs for the optional flow and rain sensors. The board has plenty of space below the screw terminals to route the wires.



electrician if the Watering System Controller is to be permanently wired to a power circuit.

Before you decide on the placement of the Controller, check that the WebMite can reach your WiFi network at that location. As described later, you can do this by loading the firmware onto the WebMite and powering it with a 5V USB power bank or portable computer. If you can call up its web page from that location, you are good to go.

24V AC power source

24V AC is the standard power supply for domestic watering system controllers and if you are replacing an

existing controller, it might already be available.

We mounted the prototype Watering System Controller next to the house fusebox, which had a mains GPO socket inside, then used a 24V AC plug pack to power the Controller. This is the best and most economical solution; suitable plug packs are inexpensive and easy to find. Typical examples are Jaycar MP3032 and Altronics M9379A.

If you want to power the Controller from the mains, you will need a qualified electrician to run the cables and connect them to a power circuit. This is expensive, so we do not recommend it, but if you must, the PCB

has space for a small toroidal transformer at the top, such as Jaycar Cat MT2112 or Vigortronix VTX-146-030-212.

The Controller does not have facilities for terminating and fusing the transformer primary as that would be done by the electrician.

Construction

The Controller is built on a 132 × 152mm double-sided PCB coded 15110231, with the components mounted as shown in Fig.3. It is available from the PE PCB Service

Before you start, you need to decide what options you will include or exclude. The first is the number of

valves to control. Most reticulation schemes only need two, three or four sprinkler valves, but the Controller can control up to eight to accommodate large layouts. Most constructors will choose one of the options mentioned in the parts list: four, six or eight valves.

The two sensor inputs are also optional. You can omit the associated components if you do not plan to install a rain or flow sensor. Still, you might want to fit them in case you decide to use them later.

The two LEDs and tactile switches are also optional. They are helpful if you are fault-finding or setting up the Controller, but they will be of little use once the box is sealed. You could mount them on the front panel using waterproof LEDs and switches with flying leads, where they would be much more useful.

We mounted a waterproof power switch and status LED on the front

panel of our prototype. If you see the LED flashing, you can visit the Controller's web page to determine the cause.

All the components are through-hole types, so construction should be easy and there are no particular tricks. Follow Fig.3 and the silk-screened text on the PCB and start with the low profile components, working towards the taller components.

The WebMite (Pico W) can be soldered directly to the PCB, but we strongly recommend using sockets on the PCB and pin headers on the WebMite to make it a plug-in device. That way, you can easily remove it for testing and fault-finding.

Loading the firmware

Before you plug the WebMite into the board, load the firmware via the USB port on a computer (Windows, Mac or Linux). Starting with

a factory-fresh Raspberry Pi Pico W, you first need to load the WebMite firmware (MMBasic). The process is described in detail in the WebMite User Manual, but we will summarise it here:

- 1. While holding down the white button on the top of the Pico, plug it into your computer. The Pico should appear as a pseudo USB drive on your computer.
- 2. Locate the WebMite firmware (with a name like WebMiteV5.07.07.uf2) and drag and drop that into the USB drive.
- 3. When it finishes copying, the WebMite will reboot and reconnect to your PC as a serial port over USB. The green LED on the top of the Pico W should slowly flash.
- 4. Determine the name of the serial port used by the WebMite (ie, COM12), then use Tera Term (<http://tera-term.en.lo4d.com>) to connect to that port.

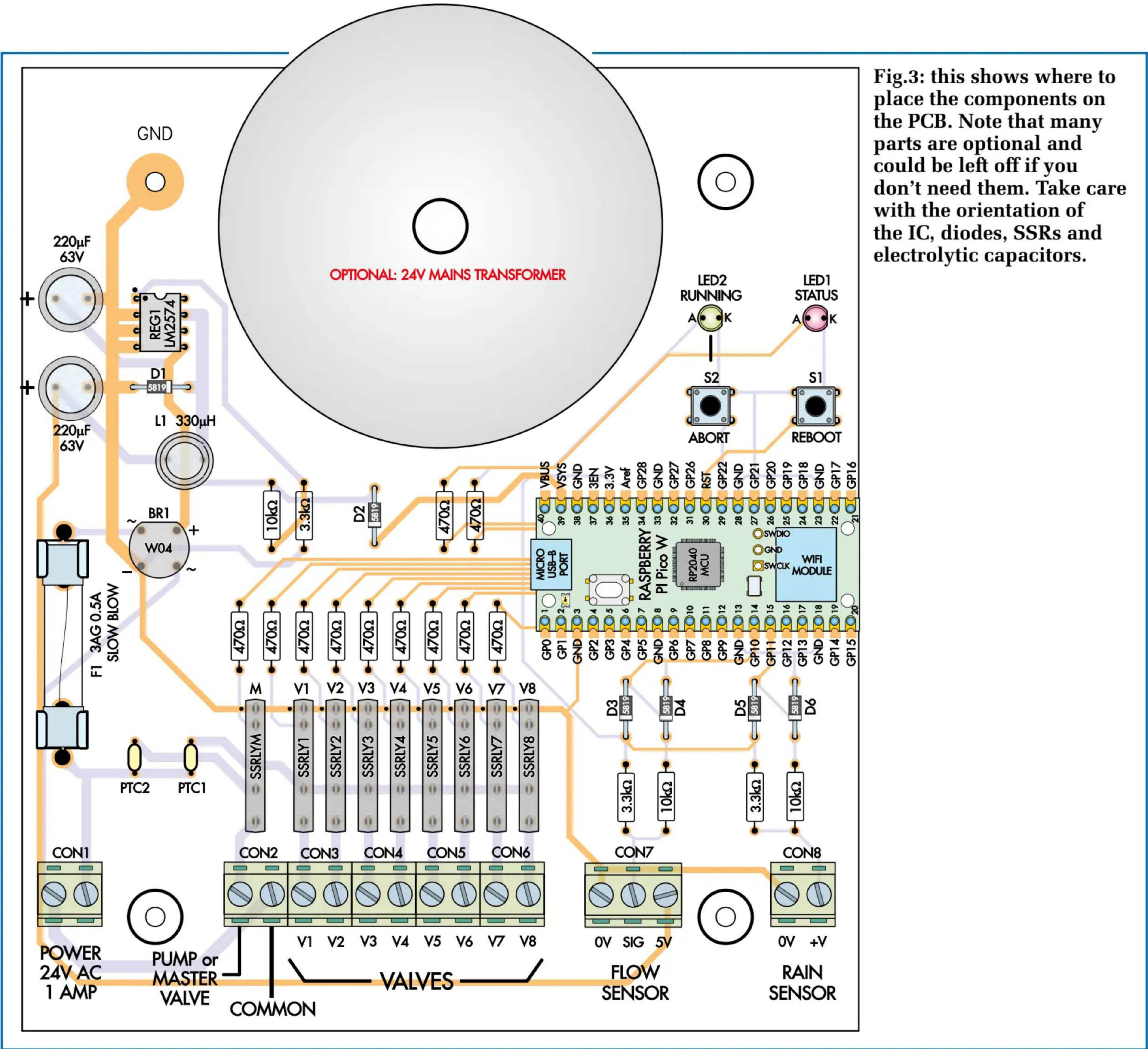


Fig.3: this shows where to place the components on the PCB. Note that many parts are optional and could be left off if you don't need them. Take care with the orientation of the IC, diodes, SSRs and electrolytic capacitors.

5. Press return/Enter and you should see the MMBasic command prompt (the > character).

Next, you must set the WebMite options for accessing the WiFi network. These are entered at the command prompt and each will cause the WebMite to reboot, so you will need to reconnect the Tera Term session after each (the double quotes are required for the SSID and password values):

```
OPTION WIFI "SSID", "password"
OPTION TCP SERVER PORT 80
OPTION TELNET CONSOLE ON
```

The following command will show the IP address that your router has allocated to the WebMite:

```
PRINT MM.INFO(IP ADDRESS)
```

Now you need to copy some files to the WebMite's internal file system. The best way to do this is by using TFTP on your computer. Follow these instructions for Windows: siliconchip.au/link/abmf

The files are in the download for the Watering System Controller from the August 2024 page of the PE website: <https://bit.ly/pe-downloads>

They are named "retic.bas", "config.html", "index.html" and "setup.html". Use the following TFTP commands to transfer the files (ipaddr is the IP address of the WebMite):

```
TFTP -i ipaddr PUT retic.bas
TFTP -i ipaddr PUT config.html
TFTP -i ipaddr PUT index.html
TFTP -i ipaddr PUT setup.html
```

You can check the files have copied correctly by using the command FILES at the MMBasic prompt. That will list the files in the internal file system.

Now you can run the BASIC program using the following command and the Watering System Controller should start up:

```
RUN "retic.bas"
```

When the program starts, it will print a series of status messages on the console. The program does this whenever it takes some action; it is useful for debugging and understanding what is happening.

You can now fire up your web browser and enter in the address field the IP address of the WebMite that you determined earlier. When you press Enter, you should then see a web page similar to Screen 1 (without the configuration data entered).

Note that it's possible to skip all the steps above apart from setting the WiFi options and running the program by instead loading the all-in-one "WaterCtrl.uf2" file that's

Parts List – Watering System Controller

- 1 double-sided PCB coded 15110231, 132 × 152mm
- 1 Raspberry Pi Pico W microcontroller module (MOD1)
- 1 RITEC RP1285BF 186 × 146 × 75mm sealed enclosure
[Altronics [H0310F](#), DigiKey [164-RP1285BF-ND](#), Mouser [546-RP1285BF](#)]
- 1 330µH 0.5A bobbin-style inductor (L1) [Altronics [L6227](#)]
- 1 IXYS CPC1966Y or CPC1965Y solid-state relay (SSRLYM)
[Mouser [849-CPC1966Y](#) or [849-CPC1965Y](#)]
- 2 500mA hold current, 1A trip PPTC resettable fuses (PTC1-2)
[Altronics [R4550A](#) or Bourns [MF-RX050/72-AP](#)]
- 2 PCB-mounting momentary tactile switches (S1, S2) [Altronics [S1120](#)]
- 2 two-way 5/5.08mm 45° PCB-mounting terminal blocks (CON1, CON2)
[Altronics [P2044A](#)]
- 2 20-pin headers, 2.54mm pitch (for MOD1)
- 2 20-pin header sockets, 2.54mm pitch (for MOD1)
- 2 3AG PCB-mounting fuse clips (F1) [Altronics [S5980](#)]
- 1 3AG slow-blow 500mA fuse (F1)
- 4 No.4 × 6mm panhead self-tapping screws

Semiconductors

- 1 LM2574(Y)N-ADJ buck regulator, DIP-8 (REG1)
- 1 W04 400V 1.2A bridge rectifier (BR1) [Altronics [Z0073](#) or [Z0073A](#)]
- 2 1N5819 40V 1A schottky diodes (D1, D2)
- 1 3mm red LED (LED1)
- 1 3mm green LED (LED2)

Capacitors

- 2 220µF 63V 105°C radial electrolytic caps

Resistors (all ¼W 5% axial)

- 1 10kΩ 1 3.3kΩ 3 470Ω

Value	4-band code	5-band code
10kΩ		
3.3kΩ		
470Ω		

Extra parts for a four-, six- or eight-valve controller

- 4, 6 or 8 IXYS CPC1966Y or CPC1965Y solid-state relays (SSRLY1-SSRLY8)
[Mouser [849-CPC1966Y](#) or [849-CPC1965Y](#)]
- 4, 6 or 8 470Ω ¼W 5% axial resistors
- 2, 3 or 4 two-way 5/5.08mm 45° PCB-mounting terminal blocks (CON3-CON6) [Altronics [P2044A](#)]

Extra parts for the flow sensor

- 2 1N5819 40V 1A schottky diodes (D3, D4)
- 1 10kΩ ¼W 5% axial resistor
- 1 3.3kΩ ¼W 5% axial resistor
- 1 three-way 5/5.08mm 45° PCB-mounting terminal block (CON7)
[Altronics [P2045A](#)]
- 1 5V-powered flow sensor, TTL output
[Valves Direct siliconchip.au/link/abmg]

Extra parts for the rain sensor

- 2 1N5819 40V 1A schottky diodes (D5, D6)
- 1 10kΩ ¼W 5% axial resistor
- 1 3.3kΩ ¼W 5% axial resistor
- 1 two-way 5/5.08mm 45° PCB-mounting terminal block (CON8)
[Altronics [P2044A](#)]
- 1 rain sensor with NC contacts [Valves Direct siliconchip.au/link/abmh]

included in the download package.

Now you can install the WebMite on the Controller PCB, place the Controller in its final location and attach the case lid. You can then set up the Controller as described earlier by accessing it via WiFi, configuring each watering schedule, completing the basic setup section to enter your location etc.

Fault-finding

If the Controller does not work as expected, first check the status LED. If it is off or flashing, that means a

fault has been detected; if it is permanently illuminated, that means that everything is working correctly (as far as the program is concerned). While the Controller is starting up, the LED will turn off or flash for a while, but if it is still flashing after a minute or two, something is wrong.

If the LED indicates an error, log into the Controller web page and check for a message in red at the bottom. This could be an incorrect configuration, inability to access the internet, excessive water flow

Modifying the BASIC program

While the Controller is configured via its web pages, you can also change some minor settings by editing the BASIC program. To do this, use Telnet to connect to the WebMite's console (eg, using Tera Term), then press CTRL-C to interrupt the running program.

Enter EDIT at the command prompt to run the editor. Scrolling down, you will see part of the program labelled "User changeable constants", as shown in code below. These are the parameters that you can easily change.

The status LED will flash continuously if you have not entered a location on the general setup page. To avoid this, you can turn off the warning by setting DisableLocationWarning to 1 instead of 0.

You might want to change the default time zone from AEST (+10

```
.....
' User changeable constants
.....
' set this to 1 to disable the location warning
Const DisableLocationWarning = 0
' this is the default time zone if the location is not set
Const DefaultTimeZone = 10.0

' the % increase in flow rate to trigger a fault
Const UpperFtolerance = 50
' the % decrease in flow rate to trigger a fault
Const LowerFtolerance = 50

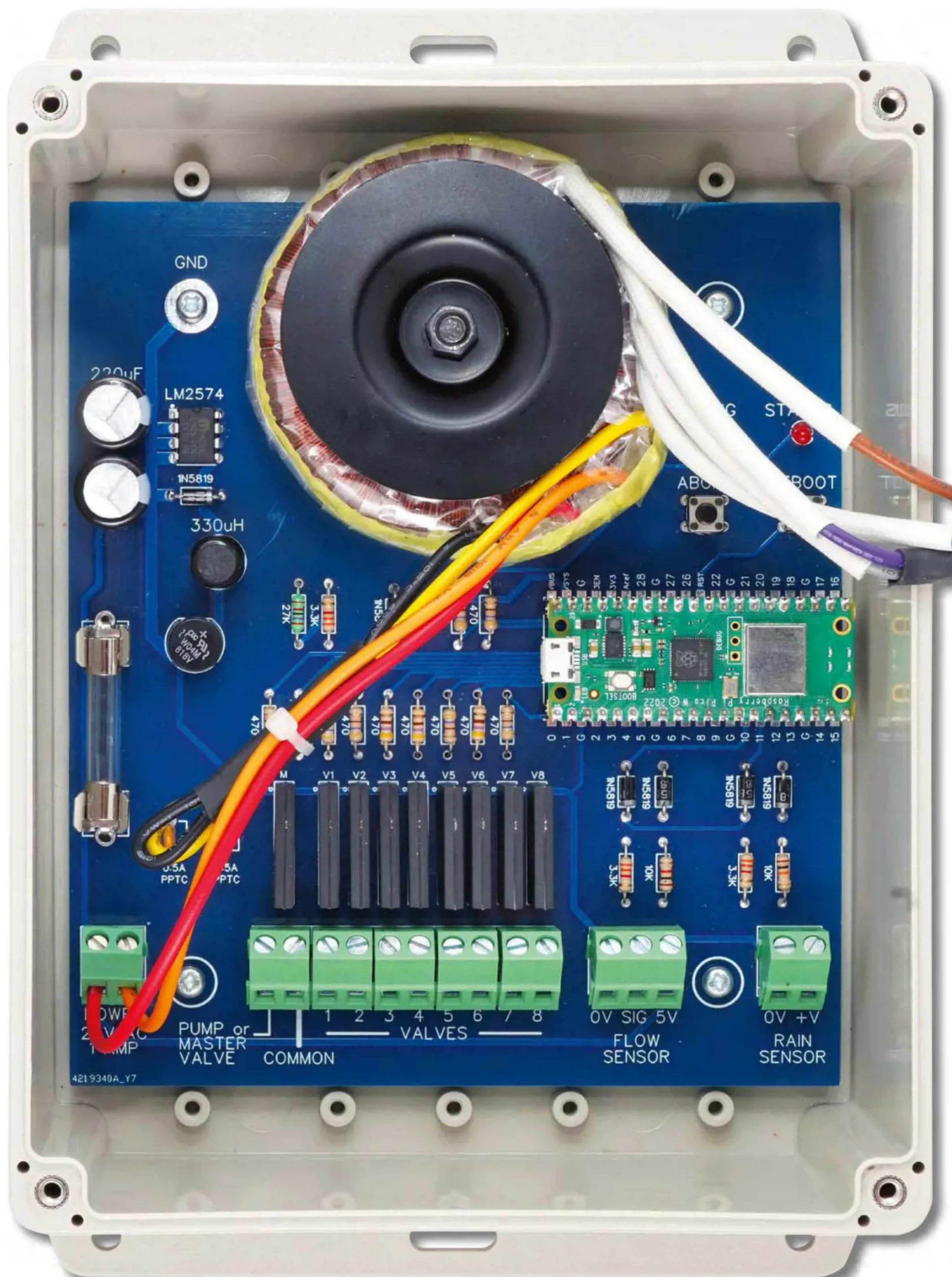
' % forecast % chance of rain required to skip a schedule
Const RainThreshold = 90

' API key for accessing Open Weather Map
' This key is shared between all users of this program and is
' limited to 60 queries/minute. This should not be a problem but
' you can easily get your own key and be independent. To do this
' goto https://openweathermap.org/ and open a free account, then
' generate a key and replace the key below with your own key.
Const OWMKey = "73cd207244614965fc5ca3646bdd10ab"
```

hours). To do this, change the line
CONST DefaultTimeZone = 10.0
to your time zone. For example,
Adelaide is 9.5. Note that the pro-
gram will not be able to correct for

daylight saving in this case.

The program will signal a fault
from the flow sensor if the value is
50% above the long-term average.
This can be changed by changing



etc. When you correct this error, the
message should vanish and the sta-
tus LED will stop flashing.

If you cannot log into the web
page or the Controller appears dead,
unplug the WebMite and plug it
directly into the USB port on your
computer. Then use Tera Term to
access the serial-over-USB console
to diagnose the problem. You might
have to press CTRL-C to terminate the
running program so you can access
the MMBasic console.

Try restarting the BASIC program
with the command RUN "retic.bas".
As the program starts, you should see
a series of messages in your terminal
emulator describing the progress.
The first will be concerned with con-
necting to your WiFi, then connect-
ing to the internet, followed by more
messages detailing the progress in
getting your location and time zone
from Open Weather Map.

The success or failure of these
should give you some pointers as to
where to look. Typical problems that
can trip you up include:

- Not programming the Pico W with the correct WebMite firmware.
- Incorrect SSID or password in the OPTION WIFI command.

Photo 2: the Controller mounted in a waterproof sealed enclosure. This shows it with an onboard 24V mains transformer, but note that it must be installed by a qualified electrician who will terminate the primary and connect it to a suitable power circuit.

the entry `Const UpperFTolerance = 50` to another value. For example, if you wanted the Controller to be much more sensitive to excessive flow, you could change the value to 20 (20%).

Similarly, for the lower tolerance (reduction in flow), change the entry `LowerFTolerance` on the line below.

We have registered an account with Open Weather Map and used the associated API key in the Watering System Controller program. Their free account provides us with everything we need. The only significant restriction is that users of the key are limited to 60 queries per minute or a million in a month.

That means that all users of this program will be using the same API key, but that should be fine as it is unlikely for 60 users to all make a call in the same minute. However, you can get your own access key if you want to be independent.

If you open an account with Open Weather Map, they will provide you with an API key. The key is a 32-character-long string of letters and numbers that acts like a password. To replace the key in the code, edit the program line starting with `"Const OWMKey ="`.

When the program gets the weather forecast from Open Weather Map, the chance of rain is returned as a percentage, with 0 representing no chance of rain and 100 indicating certainty. If a schedule is configured to skip watering on a forecast of rain, the program will do this if the chance of rain is at least 90%.

You can change this threshold to anything you want by changing the line `"Const RainThreshold = 90"`. For example, if you only want to skip watering if it is certain to rain, you can set the value to 100 (ie, 100% chance).

computer but not in the enclosure, then it could be something as simple as a wire inside the controller box draped near the WebMite's WiFi aerial, reducing its sensitivity.

If the fault appears to be with the control board, you will have to revert to traditional fault-finding procedures like checking that the power supply is working correctly and 5V is on the VSYS pin of the WebMite. Also check the component values and examine your soldering in detail.

Hopefully, your Watering System Controller will work the first time, and you can relax knowing that your plants will have all the water they need.

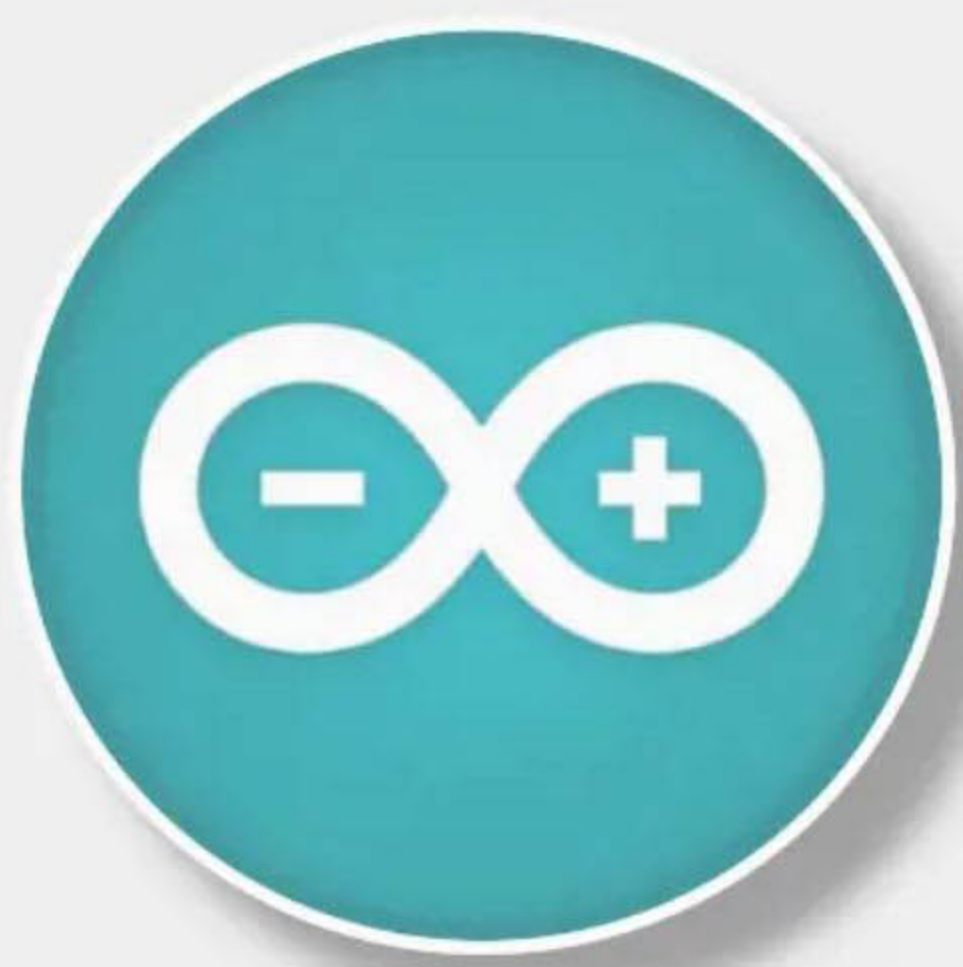
For future software updates, check the *Practical Electronics* website or the author's website at: <http://geoffg.net/retic.html>

If you enjoy using the WebMite then consider joining the Back Shed Forum (www.thebackshed.com/forum/Microcontrollers), where many enthusiastic WebMite and MMBasic users are happy to help newcomers with advice and hints.

- Your router is not configured for WPA-PSK security and DHCP.
- Your router is blocking the WebMite from reaching the internet.
- You have not set the options correctly. Use `OPTION LIST` to check them.

- You have not copied the program files to the WebMite. Use the `FILES` command to check that all four files have been copied correctly and are not zero bytes.
- If the WebMite seems to operate fine while plugged into your

Reproduced by arrangement with
SILICON CHIP magazine 2024.
www.siliconchip.com.au



Teach-In 8 CD-ROM Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

PLUS: PICs and the PICkit 3 – A beginners guide

The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE

The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.



PRICE
£8.99

Includes P&P to UK if
ordered direct from us

ORDER YOUR COPY TODAY!
JUST CALL 01202 880299 OR VISIT www.epemag.com

Circuit Surgery

Regular clinic by Ian Bell

Topics in digital signal processing – the frequency domain

We are looking at various topics related to digital signal processing (DSP). DSP covers a wide range of electronics applications where signals are manipulated, analysed, generated, stored or displayed as digital data, but originate from, and/or are output as, real-world signals for interaction with humans or other parts of the physical world. Fig.1 shows the key elements of a generic DSP system with a signal path from an analogue input via digital processing to an analogue output. This does not necessarily represent every DSP system (not all have all the parts shown) but it serves as a reference for the various subsystems we will look at.

In our discussion of sampling signals, so far we have mainly considered the signals in time domain (waveforms of voltage against time) but in DSP it is also important to understand the frequency content of signals (the frequency domain). For example, we have already discussed the Nyquist rate and the need for an antialiasing filter (see Fig.1) to remove signals which would cause errors in the sampled signal. This requirement relates to the frequencies inherently contained in sampled signals. As Matt Pulzer mentioned in his May 2024 *Editorial*, DSP is notoriously maths heavy. This month, we will cover several concepts which relate to the mathematics in DSP, but with a focus on the ideas rather than the equations. We will use one key formula and do some calculations that involve just multiplying and adding.

Spectra

To depict the frequency domain, we use graphs of signal strength versus frequency – the signal's spectrum. We also plot parameters such as the gain of a circuit against frequency as a frequency-response graph. Looking at frequency responses and signal spectra together shows us the

effect circuits or signal processing will have on a signal and helps us understand design requirements. The frequency axis of signal spectra and response plots may be linear or logarithmic, with logarithmic axes often used when a very wide range of frequencies is considered.

Signal strength on spectra may be plotted with various parameters such as voltage, current, power or power density. Although theoretical spectra can be continuous functions, the practicalities of measurement and simulation usually result in a spectrum that is effectively a histogram comprising frequency ranges or 'bins' for which the signal strength is plotted. It follows that if we change the width of the bins the signal level in each bin may also change. Using spectral density avoids this problem, however, when spectra are drawn simply to show the general shape of the signals at different frequencies, rather than provide specific numerical data/measurements, the units and bin ranges are often less important.

The sinewave has the simplest spectrum with a single peak at just one frequency (see Fig.2). Other simple periodic waveforms, such as square waves, have spectra with peaks with specific relative magnitudes at specific sets of individual frequencies (see Fig.3). Non-periodic, meaningful waveforms, such as voice signals, contain a wide range of different frequencies, but with stronger components at some frequencies than others, and a potentially complex variation of signal strength with frequency.

Often when discussing signal processing we want to depict the spectrum of a generic non-periodic signal, such as speech or music. Fig.4 shows a spectrum of such a signal which has content from DC to 2kHz. Fig.5 shows a spectrum of a signal with no DC (zero amplitude at zero frequency) but also with content in the range up to 2kHz. Spectral shapes

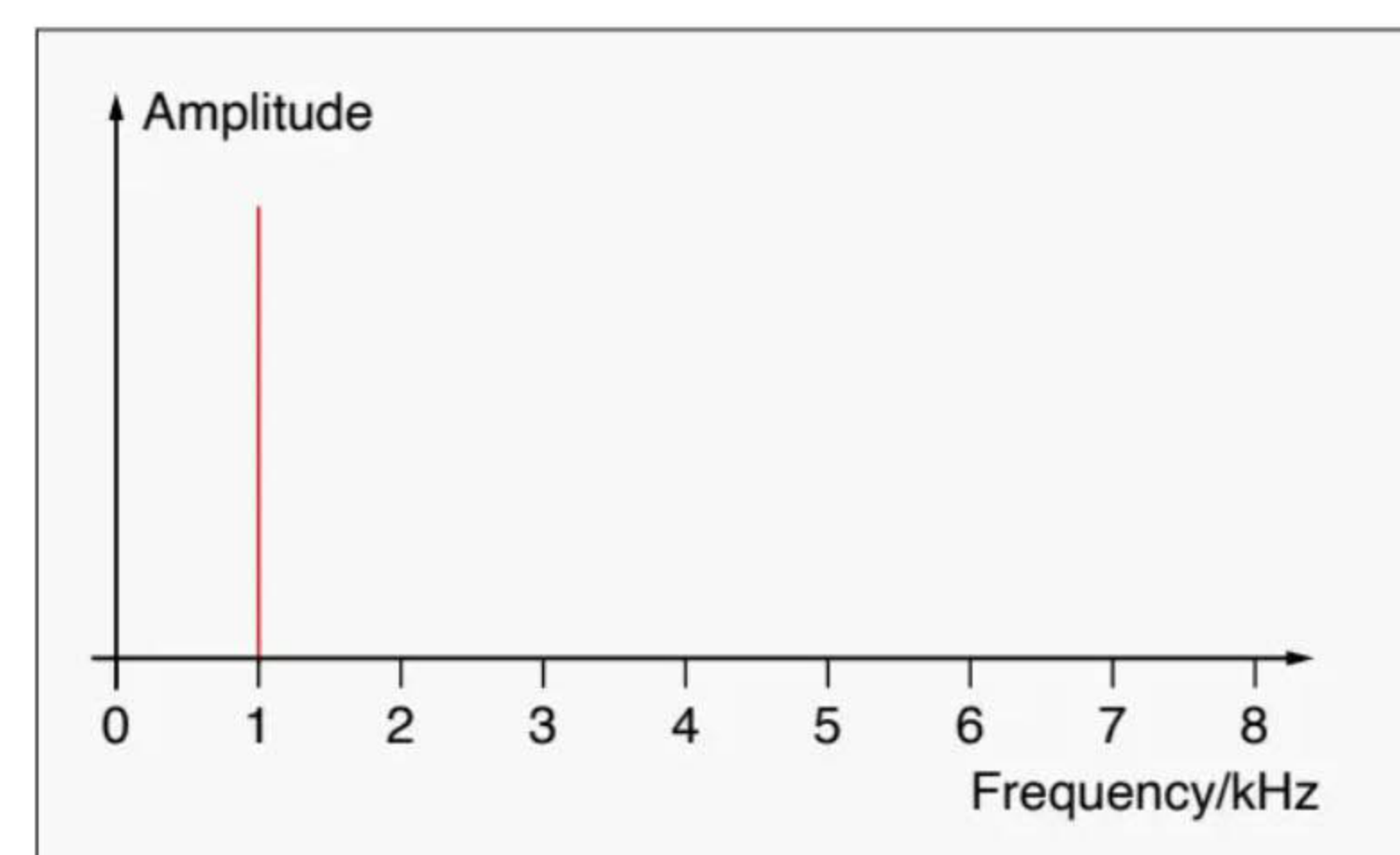


Fig.2. Spectrum of a 1kHz sinewave.

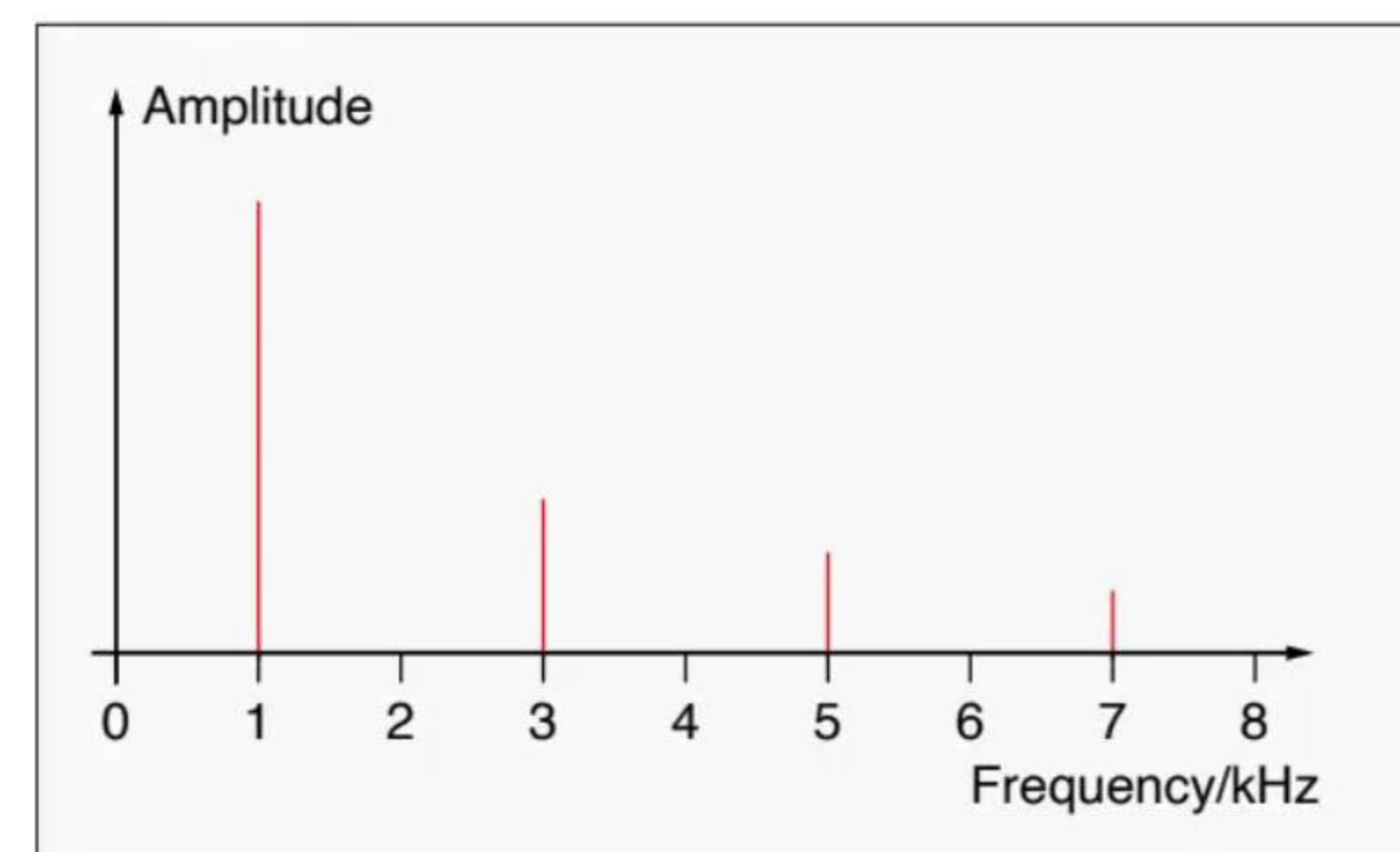


Fig.3. Spectrum of a 1kHz square wave.

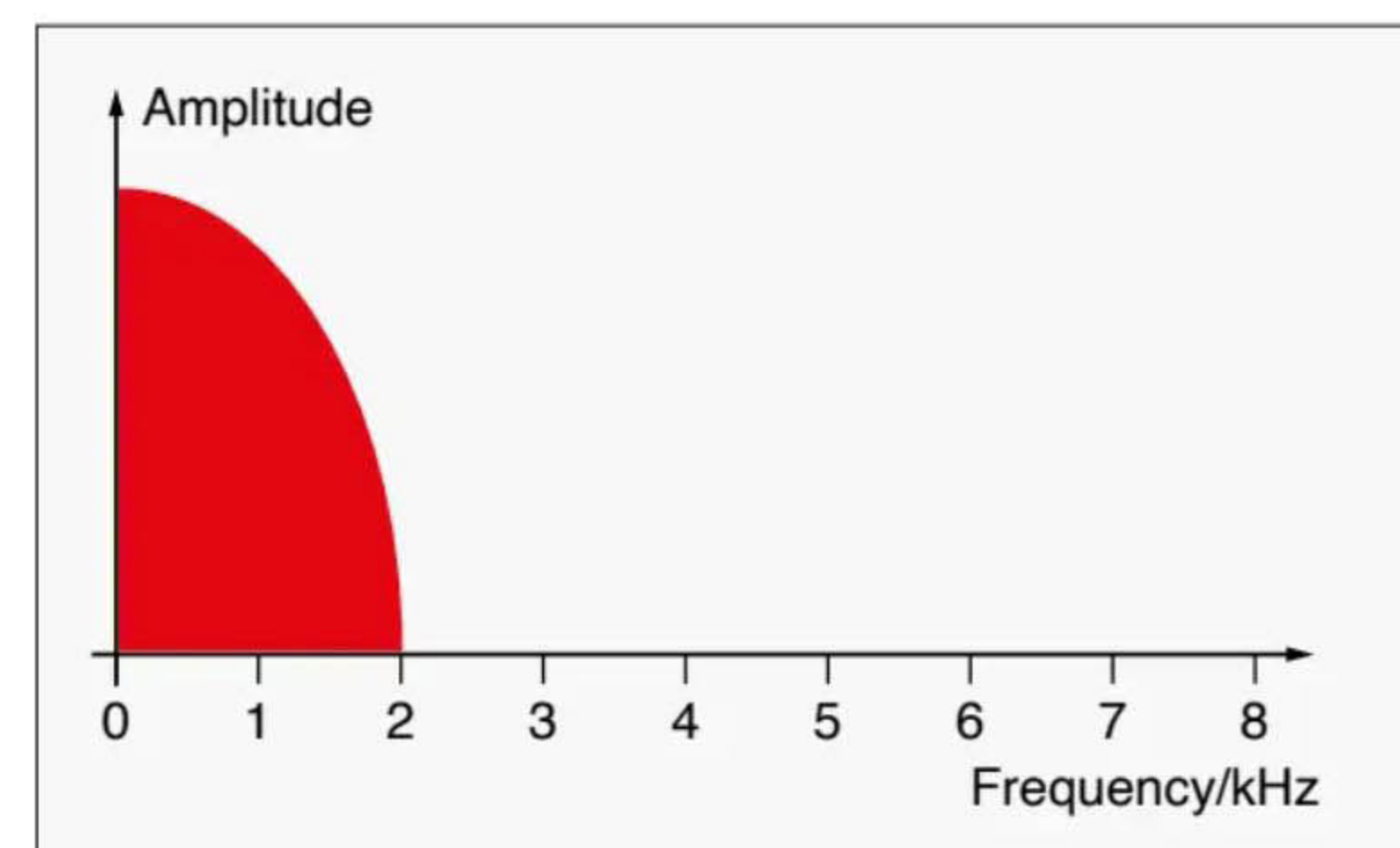


Fig.4. Spectrum of a generic non-period waveform with DC.

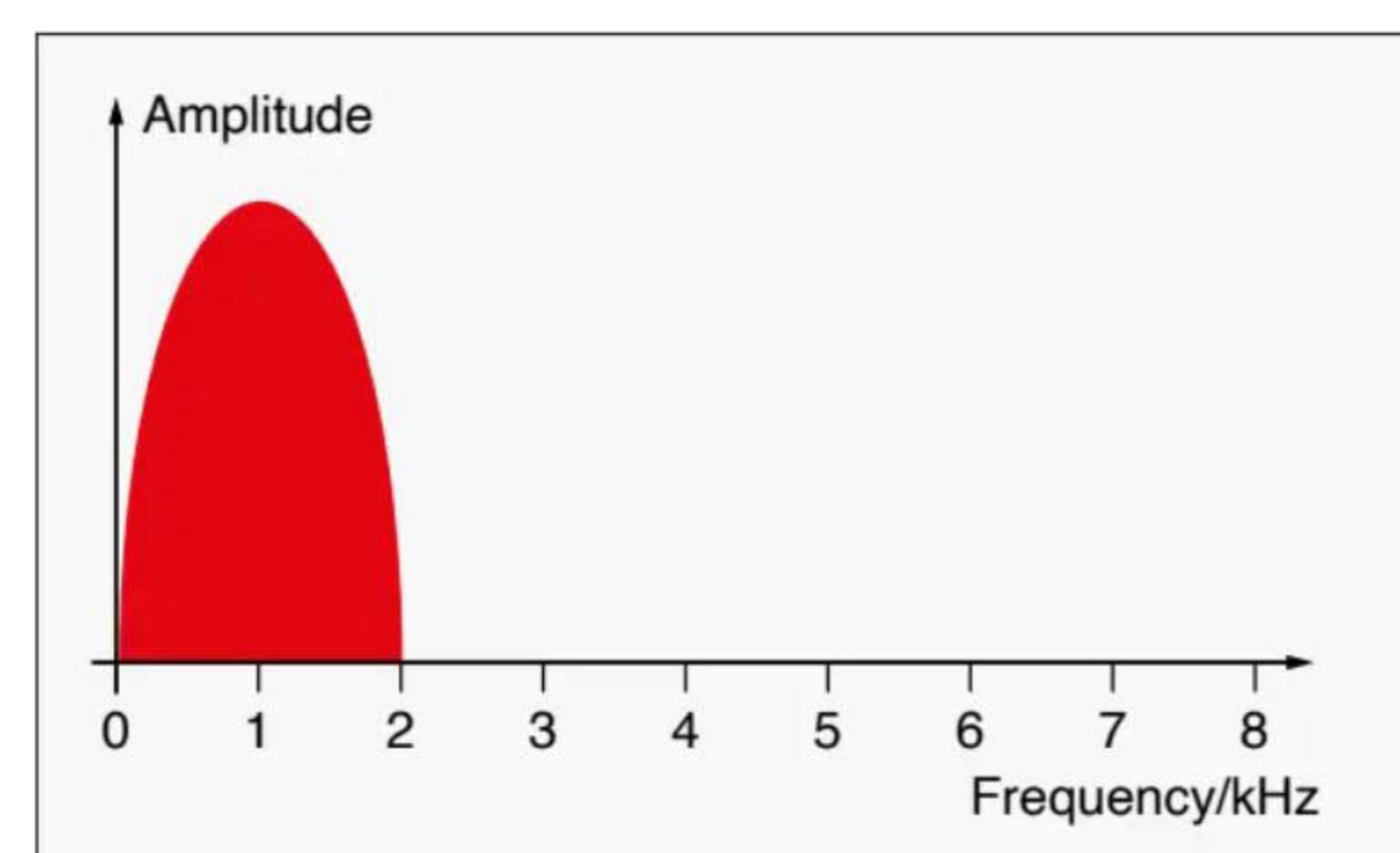


Fig.5. Spectrum of a generic non-period waveform with no DC.

such as these are commonly used when discussing signal processing in the frequency domain. The spectra in such figures are often drawn on linear axes which can go down to zero (DC) – log frequency axes cannot show zero frequency (minus infinity on a log scale).

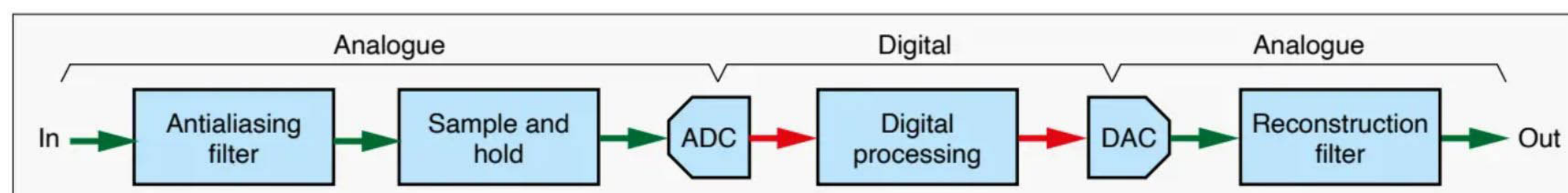


Fig.1. Generic DSP system structure.

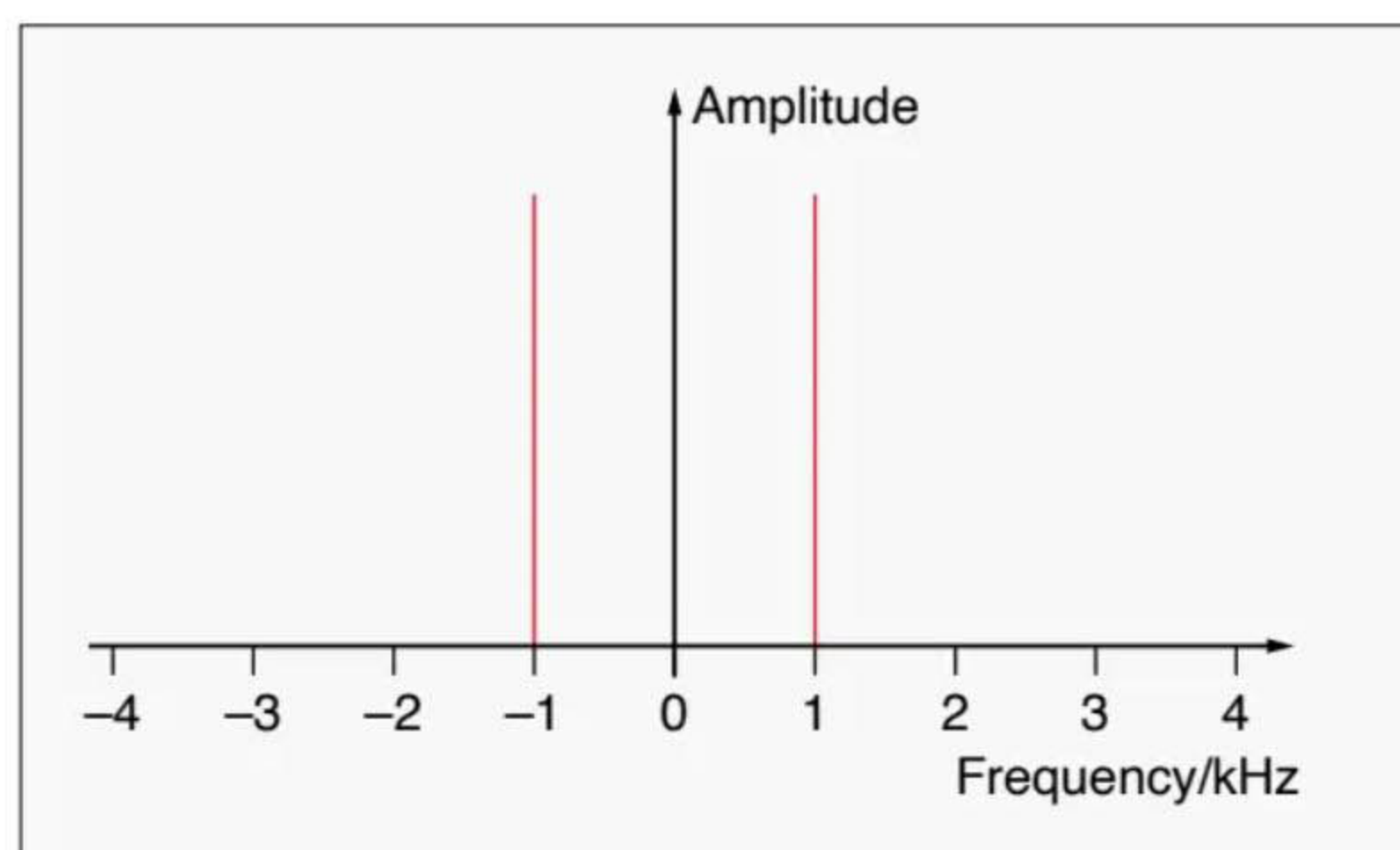


Fig.6. Spectrum for a 1kHz sinewave with negative frequencies shown (same signal as shown in Fig.2).

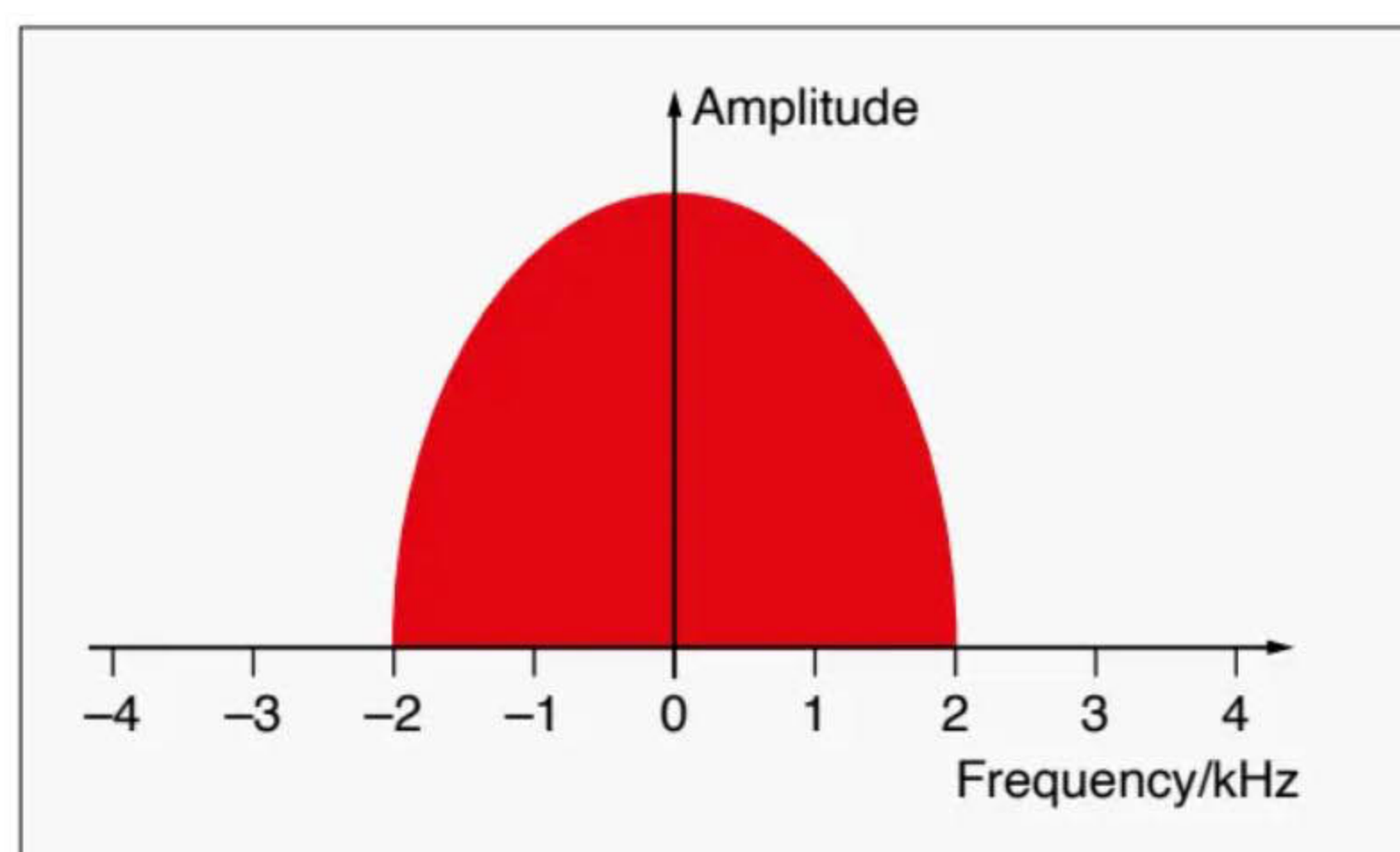


Fig.7. Spectrum of a generic non-period waveform with DC with negative frequencies shown (same signal as shown in Fig.4).

The mathematics behind the idea that a signal can be represented as a sum of sinewaves (this is effectively what a spectrum shows) was first proposed by the French mathematician Joseph Fourier (1768 – 1830). The set of peaks in a periodic waveform is called its Fourier series, and the more general mathematical process for converting from a time domain to a frequency domain representation of a signal is called the Fourier Transform (and the reverse or inverse Fourier Transform goes the other way). For the sampled signals used in DSP the basic Fourier Transform is replaced by the Discrete Fourier Transform (DFT). There are several methods of calculating the DFT from the signal sample data, but one method, the Fast Fourier Transform (FFT), as its name suggests, is much more computationally effective than the others, so FFT is the usual approach. To see the spectrum of a waveform in LTspice you use View > FFT from the right-click menu on the waveform.

A consequence of the Fourier Transform is that the spectrum of a signal contains negative frequencies. This concept may be hard grasp intuitively – it is a result of the *mathematical* representation of the spectrum rather than being part of our direct perception of the world. It is related to the need to represent signals and circuit responses using ‘complex numbers’. Amplitude alone is not sufficient to fully define a sinusoidal signal of a given frequency – we also need to know phase (offset in time). Phase and amplitude are not the most convenient way of representing the signal for mathematical analysis. An alternative, also with two values, is to represent the

signal using complex numbers. The two parts of a complex number are known as the ‘real’ and ‘imaginary’ parts. The term imaginary is employed because these values are multiples of the square root of -1 (given the symbol ‘ j ’ in electronics). There is no ‘normal’ number which when multiplied by itself gives -1 , but it is a very useful mathematical concept for describing things that really do happen.

For many signals the spectrum is symmetrical about zero frequency – for example, Fig.6 shows the spectrum of a 1kHz sinewave including negative frequencies and Fig.7 shows the same spectrum as Fig.4 with the negative frequencies included. The commonly seen symmetrical spectra may indicate the negative part is redundant, however, asymmetric spectra are possible; this is related to the complex number properties of the signal and occurs in cases such as the quadrature modulation mentioned in the February 2024 *Circuit Surgery*.

Impulses

An impulse is an idealised pulse. We mentioned impulses in the May 2024 *Circuit Surgery*, and it is worth discussing them in more detail because they are important in signal processing. For example, digital filters are categorised in terms of their response to impulse inputs. We can consider sampling as multiplying the continuous time signal by an impulse train (a waveform comprising a set of evenly spaced impulses). In continuous time an impulse (also called a ‘Dirac delta’) is a pulse which is of zero duration but with ‘an area under the curve’ of one. For discrete (sampled) signals an impulse is a single sample with a value of 1 at one sample point and a value of 0 for all other samples. This is called a unit sample and is related to another mathematic concept called the ‘Kronecker delta’.

Fig.8 shows a way of visualising a continuous time impulse, and a sampled signal impulse is shown in Fig.9. The top waveform (a) in Fig.8 shows a pulse that starts at time $-2T$ and ends at $+2T$, so it has a duration to 4 (in arbitrary time units). It is zero at all other times. Its amplitude is $\frac{1}{4}$ (this could be volts, but again the units are arbitrary in this discussion). The area under the pulse is simple to calculate because of its rectangular shape – it is $4 \times \frac{1}{4} = 1$. The middle waveform (b) shows a pulse half the duration and twice the amplitude of waveform (a). Its area is also 1 ($2 \times \frac{1}{2}$). Similarly, the bottom waveform is again half the duration and twice the amplitude, so the area is still 1 (1×1). We can repeat this process many times with the pulse getting shorter and greater in amplitude. Eventually the pulse duration will tend towards zero and the amplitude towards infinity, but still with

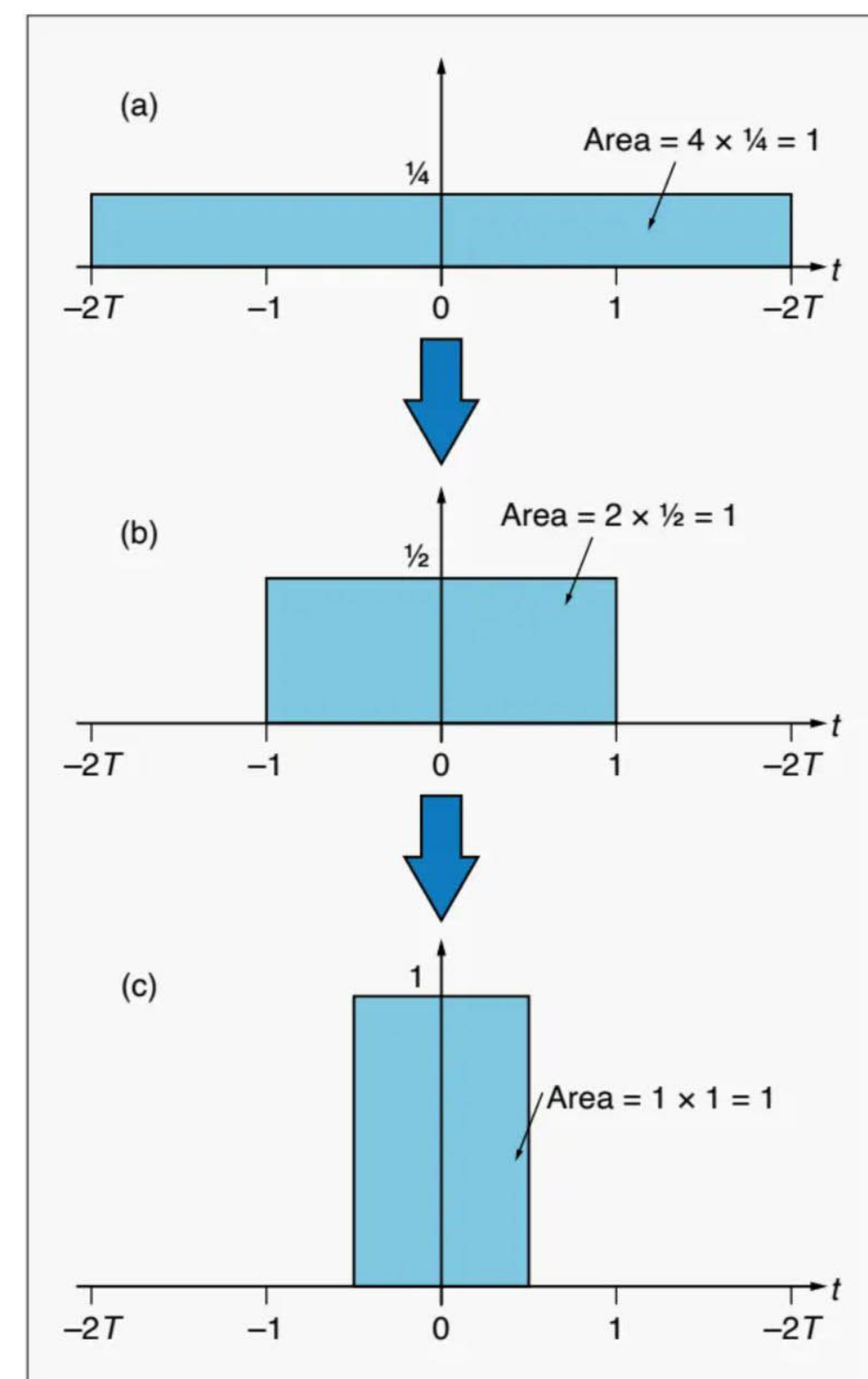


Fig.8. Visualising a continuous time impulse.

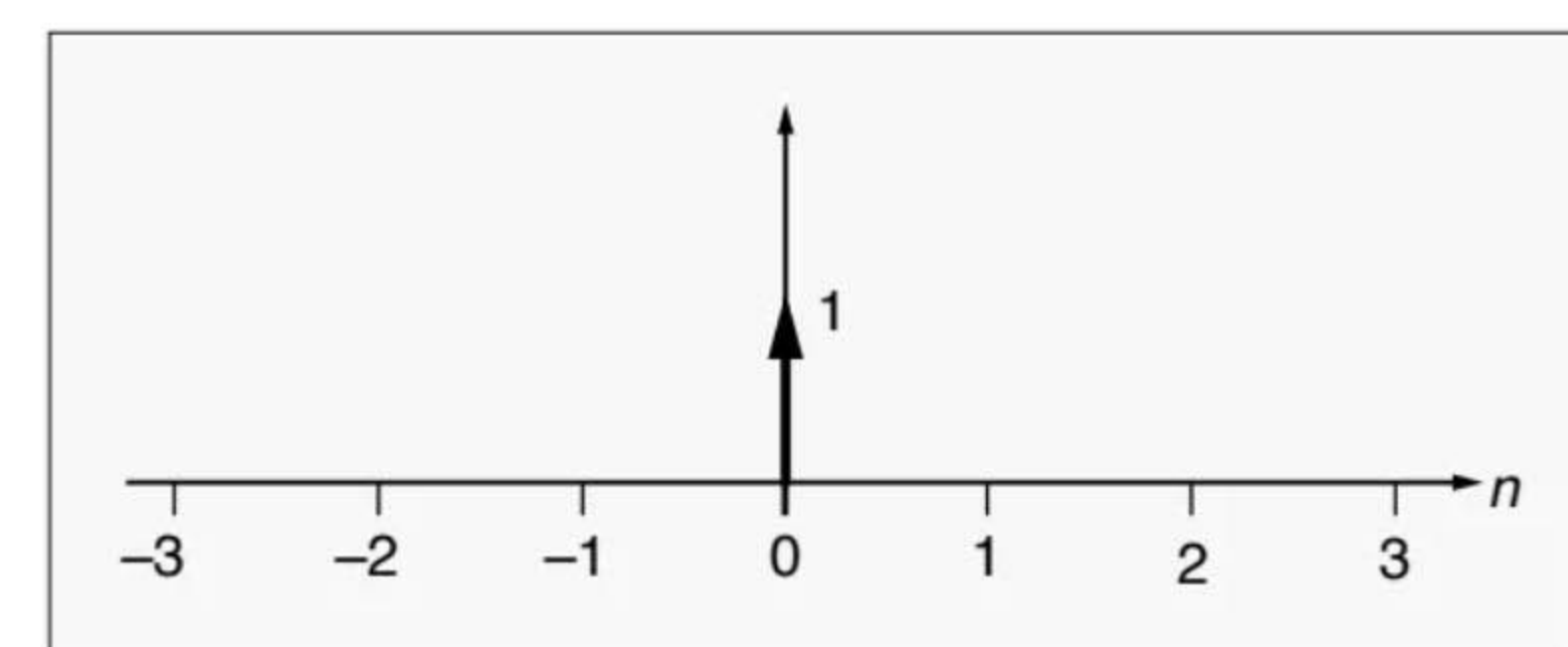


Fig.9. Unit sample (discrete time impulse).

an area of 1 – this is an impulse. The shape is not necessarily rectangular, but this representation makes it easier to show the constant area in this discussion.

Fig.10 shows a continuous time waveform, the unit impulse train used for sampling and the sampled signal. The spectrum of a unit impulse train is interesting in that it has the same shape in the frequency domain – it is also a set of evenly spaced impulses. If the time domain impulse train has amplitude 1 (unit impulse) and a period (time between impulses) of T (the sampling period) then the spectrum is an impulse train of amplitude $1/T$ spaced by $f = 1/T$, where f is the sampling frequency. The spectrum of the sampling impulse train from Fig.10 is shown in Fig.11. The arrowheads seen on the waveforms and spectrum in Fig.10 and Fig.11 are commonly used to indicate impulses. The sampling impulse train and impulse train in the spectrum are often considered to extend from minus infinity to plus infinity (in time and frequency) for theoretical work, but of course this does not happen for real signals.

Convolution

We know the spectrum of the sinewave in Fig.10 (a single frequency peak, like the spectrum in Fig.2) and we have just seen the spectrum of the impulse train from Fig.10 (Fig.11). We also know that the

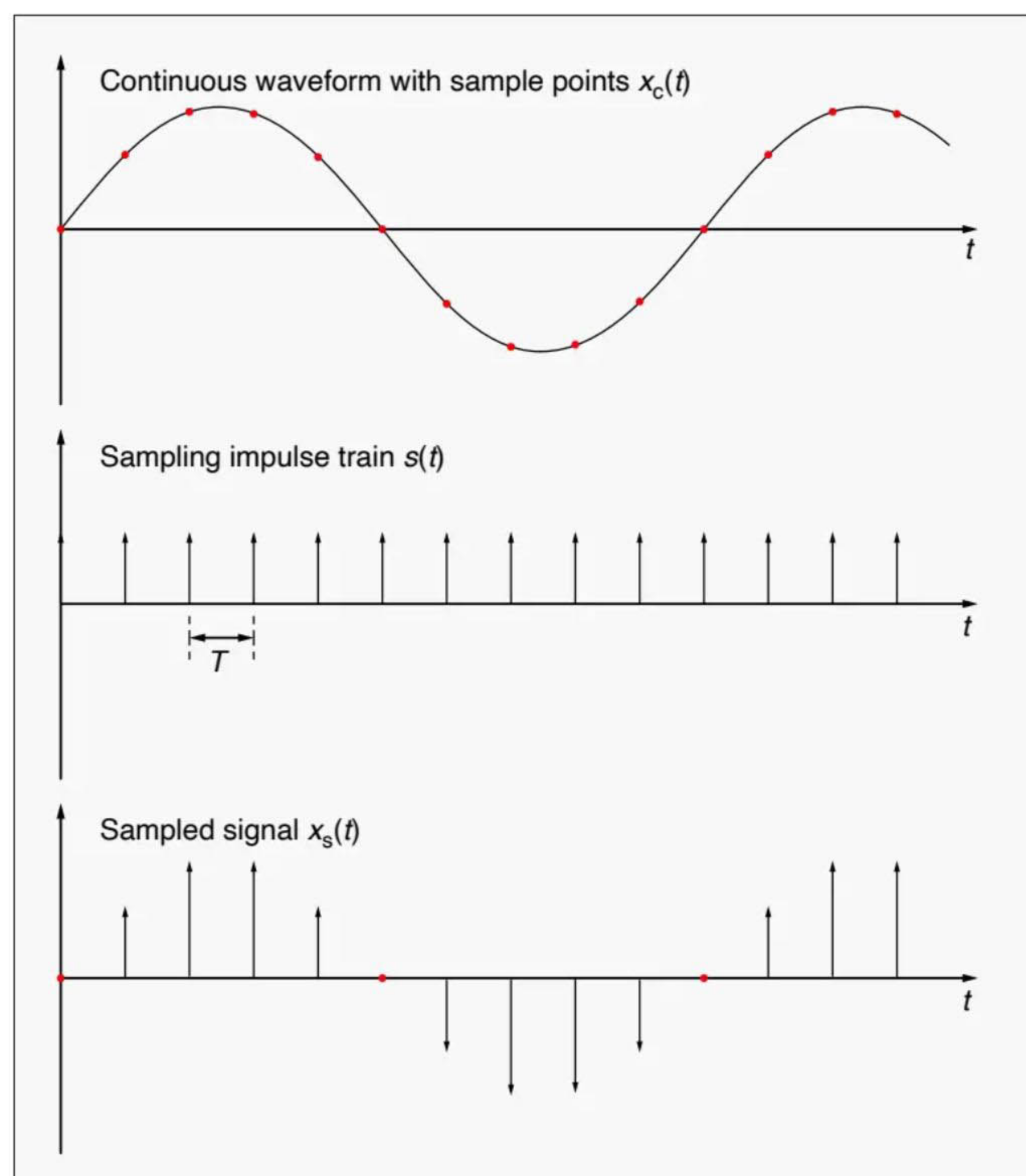


Fig.10. Time domain (waveform) view of the sampling process.

sampled signal in Fig.10 ($x_s(t)$) is related to the continuous ($x_c(t)$) and sampling ($s(t)$) signal by multiplication: $x_s(t) = x_c(t)s(t)$. The question arises as to how this relationship maps to the frequency domain, and hence what the spectrum of the sampled signal is. This leads us to an important mathematical operation called ‘convolution’ and an important theorem which states that the multiplication of two functions in time domain is equivalent to the convolution of their spectra in the frequency domain. Similarly, the convolution of two functions in time domain is equivalent to the multiplication of their spectra in frequency domain.

Unless you have studied mathematics at a fairly advanced level you may not have heard of convolution, however it is very important in a variety of areas – DSP of course – but also probability theory, image processing, artificial intelligence and various other areas of mathematics, physics and engineering. A reasonably simple example, that people might not be aware is a type of convolution, is the moving average in statistics, which is also related to the basics of digital filtering.

Convolution is an operation performed on two mathematical functions; it combines them to produce a new function. In our case (spectra related to the signals

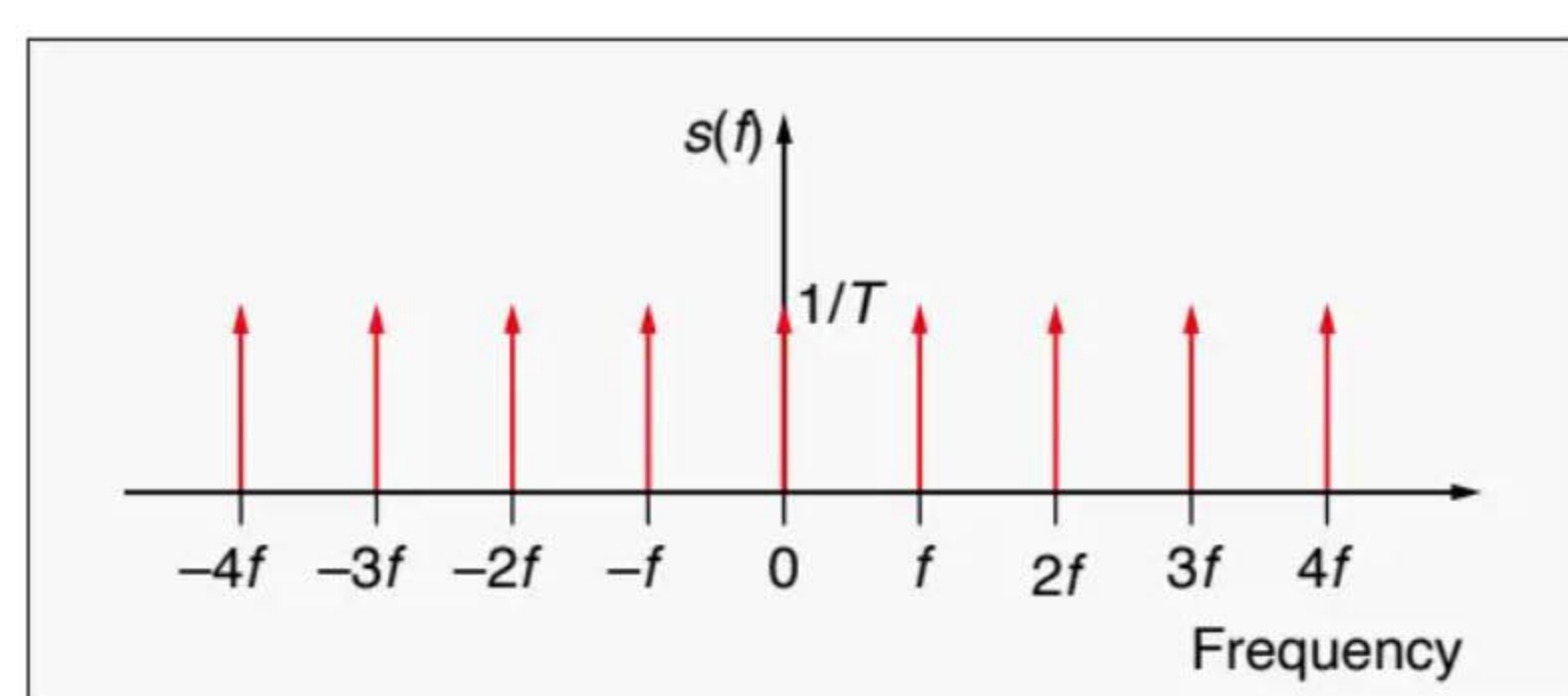


Fig.11. Spectrum of the sampling impulse train from Fig.10, where $f = 1/T$.

shown in Fig.10) the two functions are the sinewave spectrum ($x_c(f)$) (like Fig.2) and the impulse train spectrum $s(f)$ (like Fig.11). These are both functions of frequency that are convolved to find the spectrum of the sampled signal ($x_s(f)$), again a function of frequency.

Equation

Convolution can be applied to both continuous and discrete (sample-based) functions. For continuous functions its definition involves integration (calculus), but for the discrete case we replace integration with summation, which is easier to understand if you have not studied calculus. The convolution of the two discrete functions $f(n)$ and $g(n)$ is defined by the following equation:

$$f(n) * g(n) = \sum_{m=-\infty}^{\infty} f(m) g(n - m)$$

We will look at the various parts of this equation to explain what they mean. The convolution operator symbol is ‘*’, so $f(n) * g(n)$ means the convolution of functions f and g . (The fact that the symbol * is commonly used for multiplication in coding languages is a possible source of confusion.)

A function is a mathematical operation performed on one or more variables (the variables are n and m in the above). Functions are often named using a single letter (f and g in the above). For example, we could define a function called q on variable x as meaning ‘ x squared’, then we would write $q(x) = x^2$. However, functions in this context can also refer to things such as waveforms and spectra for which there is not a very simple expression such as x^2 , but for which we know or can find the value of, say, $f(n)$ for any value of n .

As noted, the above equation is the discrete version of convolution, which means the functions are of discrete variables – that is, n and m are integer (whole number) values. As mentioned above, we typically assume that waveform and spectrum data ranges from minus to plus infinity, so n and m would start at $-\infty$, go through zero, that is ... -1, -2, 0, 1, 2, 3 ... and then on up to ∞ . For real waveforms and spectra, we obviously have to use a more limited range and so typically assume all values of the function outside the range of interest are zero.

The symbol Σ (Greek capital sigma) in the convolution equation is the summation operator, which means, ‘add



Fig.12. Tetrahedral dice.

up everything in the sequence of values defined by the expression to the right of the sigma symbol ($f(m) * g(n-m)$ in this case) and the summation range definition is below and above the symbol’. Here the range is for variable m from $-\infty$ to ∞ . The value below the sigma symbol defines the start value for m via the expression $m = -\infty$ and the final value of m in the sequence is written above the sigma symbol. The value of n is fixed for a given summation in the convolution equation, but to find the complete result of the convolution (eg, a new spectrum) we need to calculate a separate summation for all values of n from $-\infty$ to ∞ . However, as noted previously we can restrict our calculations to much smaller ranges of n and m for practical applications.

Convolving dice

We will illustrate a convolution calculation in a simple context before looking at the calculation for a spectrum example. Consider throwing two dice and working out what the probability that you get a specific total value. This is easy enough to solve by writing down all the combinations, but it can also be solved by a convolution, so it is a useful example to show how the summation sequences work. We will consider two tetrahedral dice (see Fig.12) as they have just four possible values rather than the standard six – this simply reduces the amount of calculation.

For the first dice [‘die’ if you prefer] the function $f(n)$ is the probability of rolling value n , which is $1/4$ or 0.25 for $n = 1, 2, 3, 4$. Given that n can be any integer we note that for all other values of n , $f(n)$ is zero. The function $g(n)$ is the probability of rolling value n for the second dice, which has the same values as $f(n)$. The values are shown in Table 1.

The convolution $f(n) * g(n)$ is equal to the probability that the total value of the two dice will be n . Given the minimum total is 2 (rolling two 1s) and maximum is 8 (two 4s) we expect $f(n) * g(n)$ to be zero for all values except $n = 2$ to 8. We will

Table 1. Tetrahedral dice probabilities.

<i>n</i>	...	-2	-1	0	1	2	3	4	5	6	...
<i>f</i> (<i>n</i>), <i>g</i> (<i>n</i>)	0	0	0	0	0.25	0.25	0.25	0.25	0	0	0

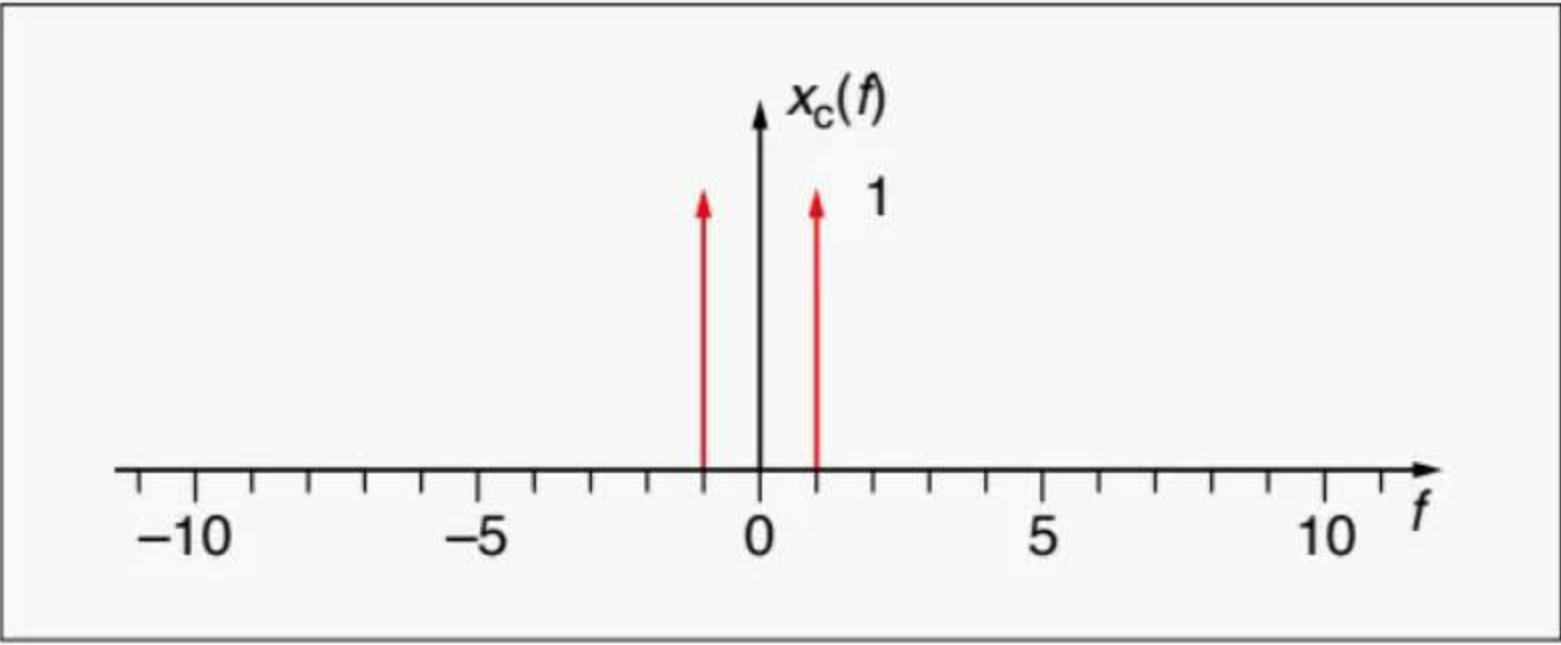


Fig.13. Spectrum of sine wave of frequency *f* = 1 which is sampled.

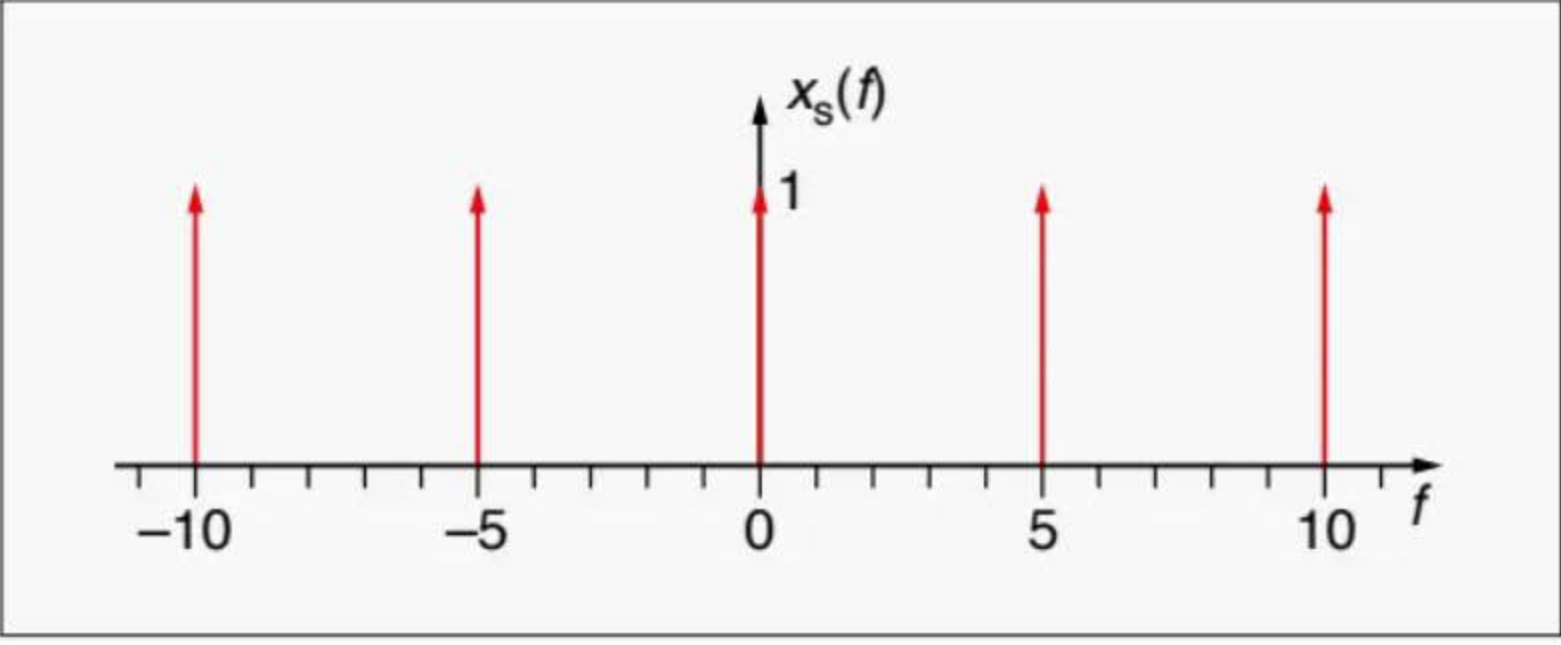


Fig.14. Spectrum of the sampling impulse train with sample frequency *f* = 5.

look at one example calculation in detail – getting a total of 6. There are 16 possible outcomes from rolling the two dice of which 3 combinations add up to 6, these are 2+4, 3+3 and 4+2, so the probability (calculated without convolution) of the total being 6 is 3/16 = 0.1875.

The convolution calculation *f*(6)**g*(6) is shown below in Table 2. This shows the summation (from the equation above) for a range of values of *m* around the points at which the results are non-zero. For all other values of *m* we know that the result will be zero so we do not need to calculate these. The convolution delivers the result *f*(6)**g*(6) = 0.0625 + 0.0625 + 0.0625 = 0.1875, the same as above.

Convolution can be described as the product of two functions where one is reflected in the y-axis (reversed) and shifted. We see in the above that in the calculation of *f*(*n*)**g*(*n*) the terms in *f*() and *g*() use different values of the variable (eg, *f*(1)*g*(5) and *f*(2)*g*(4)) – this is the ‘shifting’. Also, the variable for *g*() goes *down* in value as the summation progresses (*g*(5), *g*(4), *g*(3) and so on) – this the ‘reversal’ or ‘reflection’. If you search online you can find various animations which illustrate convolution calculations graphically to help emphasise this point – for example, <https://youtu.be/KuXjwB4LzSA>

Table 2. Convolution calculation.

<i>m</i>	...	0		1		2		3		4		5		6	...
<i>f</i> (6)* <i>g</i> (6) =	... +	<i>f</i> (0) <i>g</i> (6-0)	+	<i>f</i> (1) <i>g</i> (6-1)	+	<i>f</i> (2) <i>g</i> (6-2)	+	<i>f</i> (3) <i>g</i> (6-3)	+	<i>f</i> (4) <i>g</i> (6-4)	+	<i>f</i> (5) <i>g</i> (6-5)	+	<i>f</i> (6) <i>g</i> (6-6)	+...
<i>f</i> (6)* <i>g</i> (6) =	... +	<i>f</i> (0) <i>g</i> (6)	+	<i>f</i> (1) <i>g</i> (5)	+	<i>f</i> (2) <i>g</i> (4)	+	<i>f</i> (3) <i>g</i> (3)	+	<i>f</i> (4) <i>g</i> (2)	+	<i>f</i> (5) <i>g</i> (0)	+	<i>f</i> (6) <i>g</i> (0)	+...
<i>f</i> (6)* <i>g</i> (6) =	... +	0×0	+	0.25×0	+	0.25×0.25	+	0.25×0.25	+	0.25×0.25	+	0×0.25	+	0×0	+...
<i>f</i> (6)* <i>g</i> (6) =	... +	0	+	0	+	0.0625	+	0.0625	+	0.0625	+	0	+	0	+...

Convolution spectra

The dice example may seem a long way from DSP, but digital filters are based on calculations like these in which *f*() is the input signal and *g*() is a set of values which set the filter’s characteristics. Also, as we mentioned above, we can use convolution to calculate the spectrum of a sampled signal from the spectra of the continuous signal and sampling signal. In general, the spectra may be continuous, but by avoiding rigorous maths and choosing a simple case where we can assume discrete spectra we can provide an illustration of this.

Consider a sine wave of frequency 1 (arbitrary units, but could be kHz) and sampling rate of 5 (could be 5kHz). The sine wave spectrum is *x*_{*c*}(*f*), which corresponds to *f*() in the convolution equation and is shown in Fig.13. Its value is zero for all cases except *x*_{*c*}(-1) and *x*_{*c*}(1). The sampling signal spectrum *s*(*f*) corresponds to *g*() in the convolution equation and is shown in Fig.14. Its value is zero for all cases except *s*(0) and *s*(*n*) where *n* is a multiple of 5. In general, the amplitude of *s*(*f*) is the amplitude of the pulse divided by *T* (sampling period); however, we will just assume it is 1 for simplicity.

We can quickly calculate *x*_{*c*}(*n*)**s*(*n*) to find the sampled waveform spectrum by just considering the summation of *x*_{*c*}(*m*)*s*(*n*-*m*) for the two cases where *x*_{*c*}(*m*) is 1 (at *m* = -1 and 1) as all other cases contribute zero. This is shown in Table 3 for a small example range of *n* and the resulting spectra is shown in Fig.15.

From Table 3 and Fig.15 we see that spectrum of the sampled waveform contains the original spectrum (at *f* = -1 and 1) plus copies of the spectrum shifted from its original position centred on 0 to being centred on multiples of the sampling frequency. For example, the original spectrum is replicated centred around *f*_{*s*} to gives peaks at *f* = 4 and 6 (+1 and -1 relative to 5). Theoretically this replication occurs for all

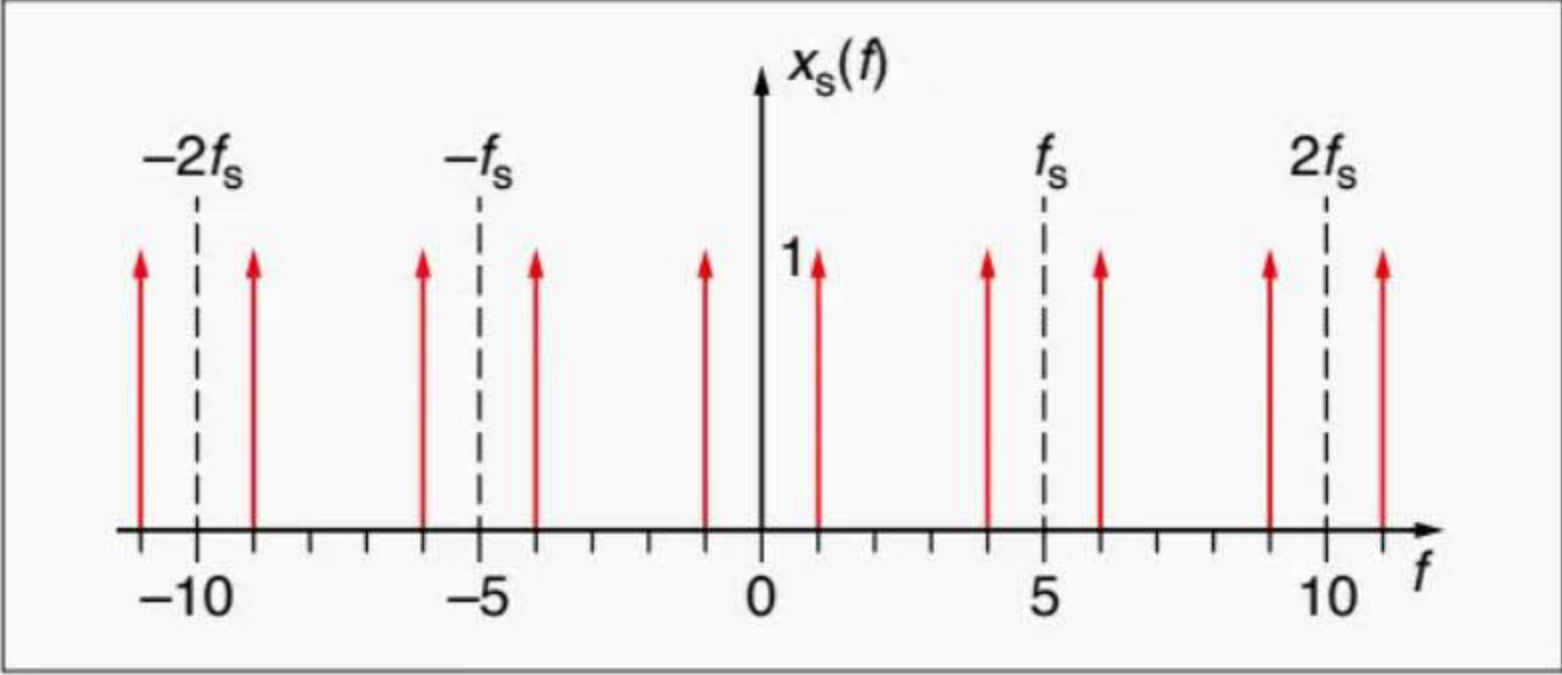


Fig.15. Spectrum of the sampled sine wave obtained by convolving the spectra in Fig.13 and Fig.14. Note the replication of the original spectrum.

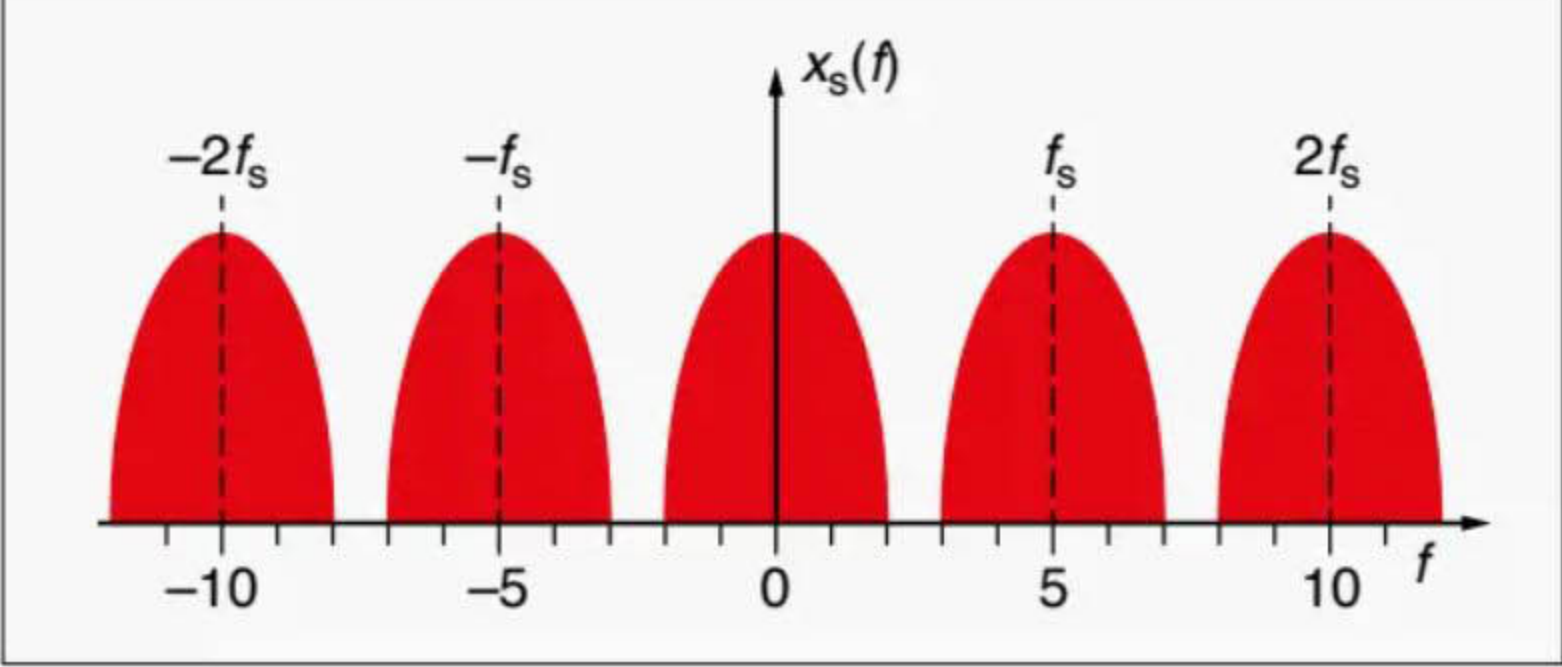


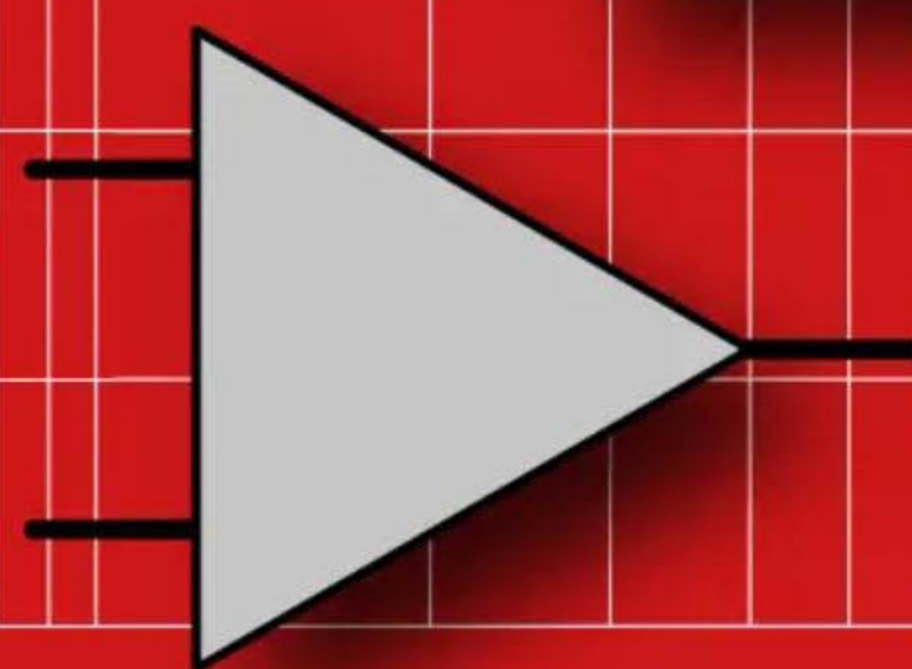
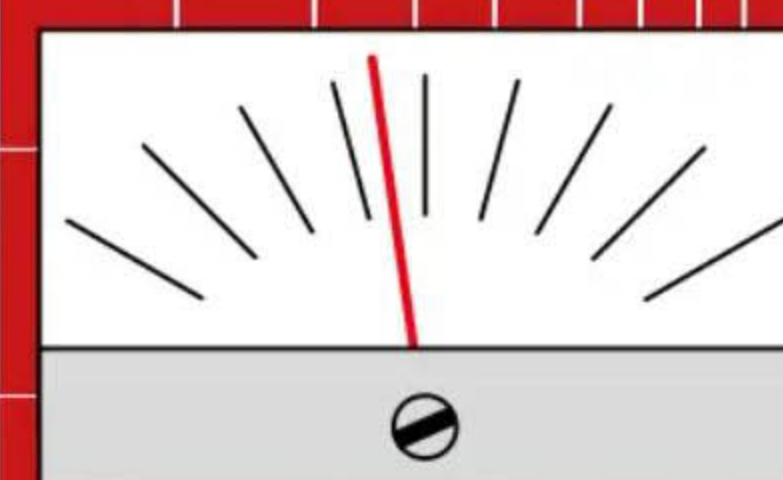
Fig.16. Spectrum of the signal similar to that in Fig.7 with a bandwidth from DC to *f* = 2 sampled at *f*_{*s*} = 5.

multiples of *f*_{*s*} to ±∞, but of course it is limited in range in real systems. In the discussion on modulation in December 2023 we saw that multiplying waveforms together produced sum and difference frequencies, and this is also what is happening here with sum and difference occurring at each multiple of the sampling frequency. A further illustration in Fig.16 shows the spectrum of a sampled version of a signal similar to that in Fig.7 where the original spectrum extends from DC to *f* = 2 and sampling frequency is again *f*_{*s*} = 5. Again, we see the replication of the spectrum at each multiple of the sampling frequency. Next month, we will look at the implications of sampled signal spectra for system design.

Table 3. Convolution to calculate sampled signal spectrum (non-zero values highlighted in blue).

	<i>m</i> = -1		<i>m</i> = 1		
<i>n</i>	<i>n</i> - <i>m</i>	<i>x</i> _{<i>c</i>} (<i>m</i>) <i>s</i> (<i>n</i> - <i>m</i>)	<i>n</i> - <i>m</i>	<i>x</i> _{<i>c</i>} (<i>m</i>) <i>s</i> (<i>n</i> - <i>m</i>)	Σ
-2	-1	0	-3	0	0
-1	0	1	-2	0	1
0	1	0	-1	0	0
1	2	0	0	1	1
2	3	0	1	0	0
3	4	0	2	0	0
4	5	1	3	0	1
5	6	0	4	0	0
6	7	0	5	1	1
7	8	0	6	0	0

AUDIO OUT



By Jake Rothman

Switching in audio equipment – Part 3

I finished off last month promising to move on to electronic switching in this month's *Audio Out*, and we will make a small start, but first there are a few odds and ends to polish off in the *mechanical* audio switching section.

'Soft' bypass

A 'soft' bypass, as opposed to a hard bypass shown in Fig.51 last month, is an arrangement that works well for internal bypassing in systems where the input is driven from a low impedance. However, it does not work well for self-contained units such as guitar pedals where the high-impedance guitar signal is loaded by the input stage at all times, pulling the impedance down. A typical place where this is used is in Hi-Fi preamplifiers to bypass the tone control – see Fig.52.

Mono switch

This is an essential test switch for music producers to listen for phase problems in stereo mixes, accomplished by mixing the left and right channels. This must be done using resistors, since it can be dangerous if the left and right channels are simply joined together. Imagine if there were two out-of-phase $10V_{rms}$ signals from two op amps (not impossible in audio) and their outputs were joined. The 'electronic smell of death' from burning epoxy would soon fill the air. The simple resistive summing circuit shown in Fig.53 leads to a halving of the voltage level ($-6dB$) if only one channel has a signal on it. It remains the same if it's the same signal on both channels, say a vocal in the middle of the mix. This means material in the (stereo) middle is louder relative to sources panned left or right when 'mono-ed'. This is fine for checking mixes, but if one really does want to listen to stereo mixes in mono it's not ideal, and compensating for this is complex. However Hi-Fi buffs with old record collections also like a mono button for playing mono records with a stereo pickup. This reduces the surface noise because some of

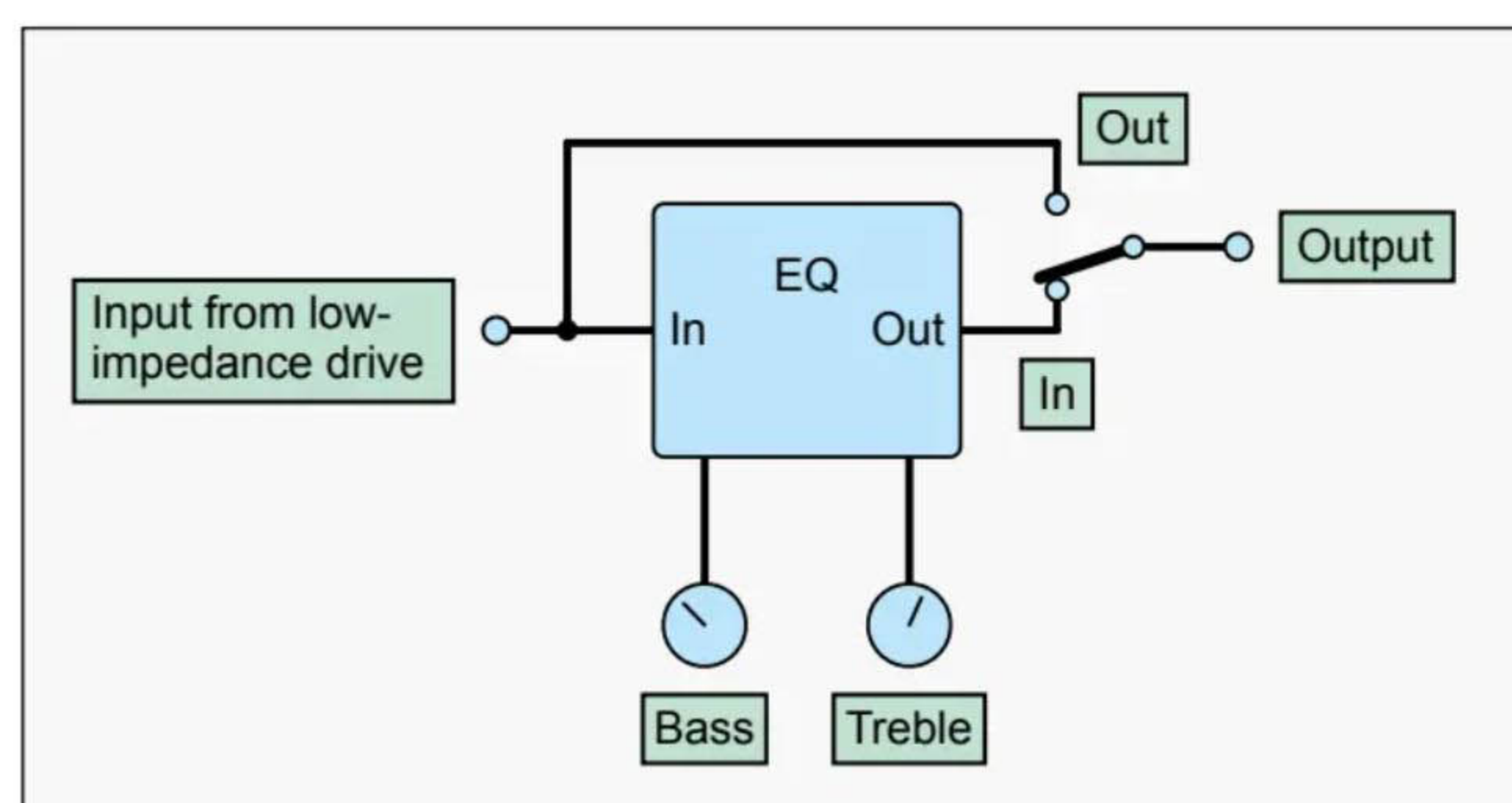


Fig.52. Alternative bypass circuit where the processor's input is always connected, such as in this Hi-Fi preamplifier configuration.

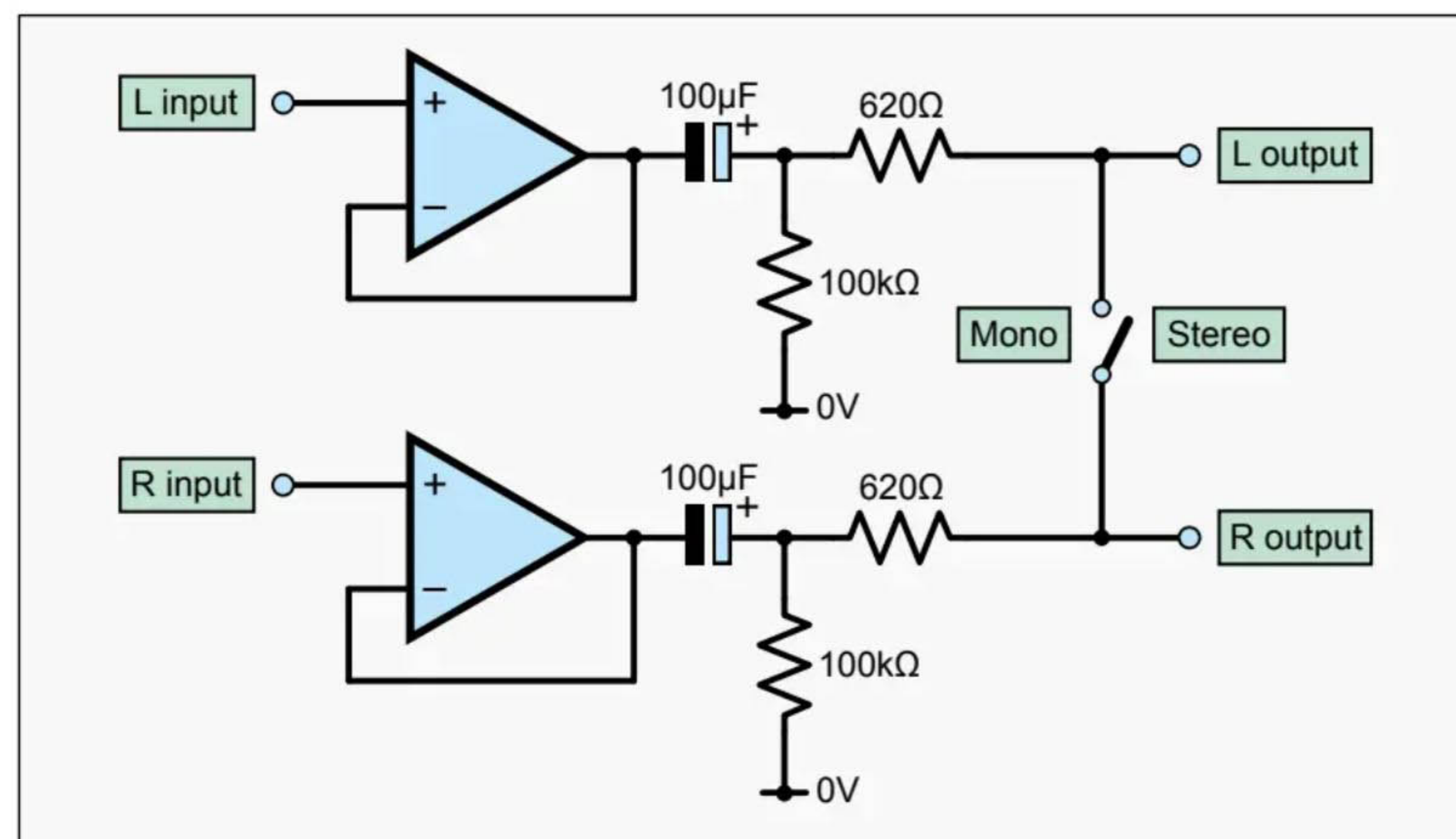


Fig.53. Traditional mono switch circuit.

it cancels out across the left and right channels. In more sophisticated units, a little bit of gain is introduced via a proper summing amplifier (Fig.54) to counteract the attenuation. A typical 'fudge value' being $+4.5dB$.

'Juggler' switch

It is sometimes necessary to switch the order of two processing blocks. A common dilemma facing music engineers is 'do we have the EQ followed by the compression or the other way round?' It sounds very different depending on which way you

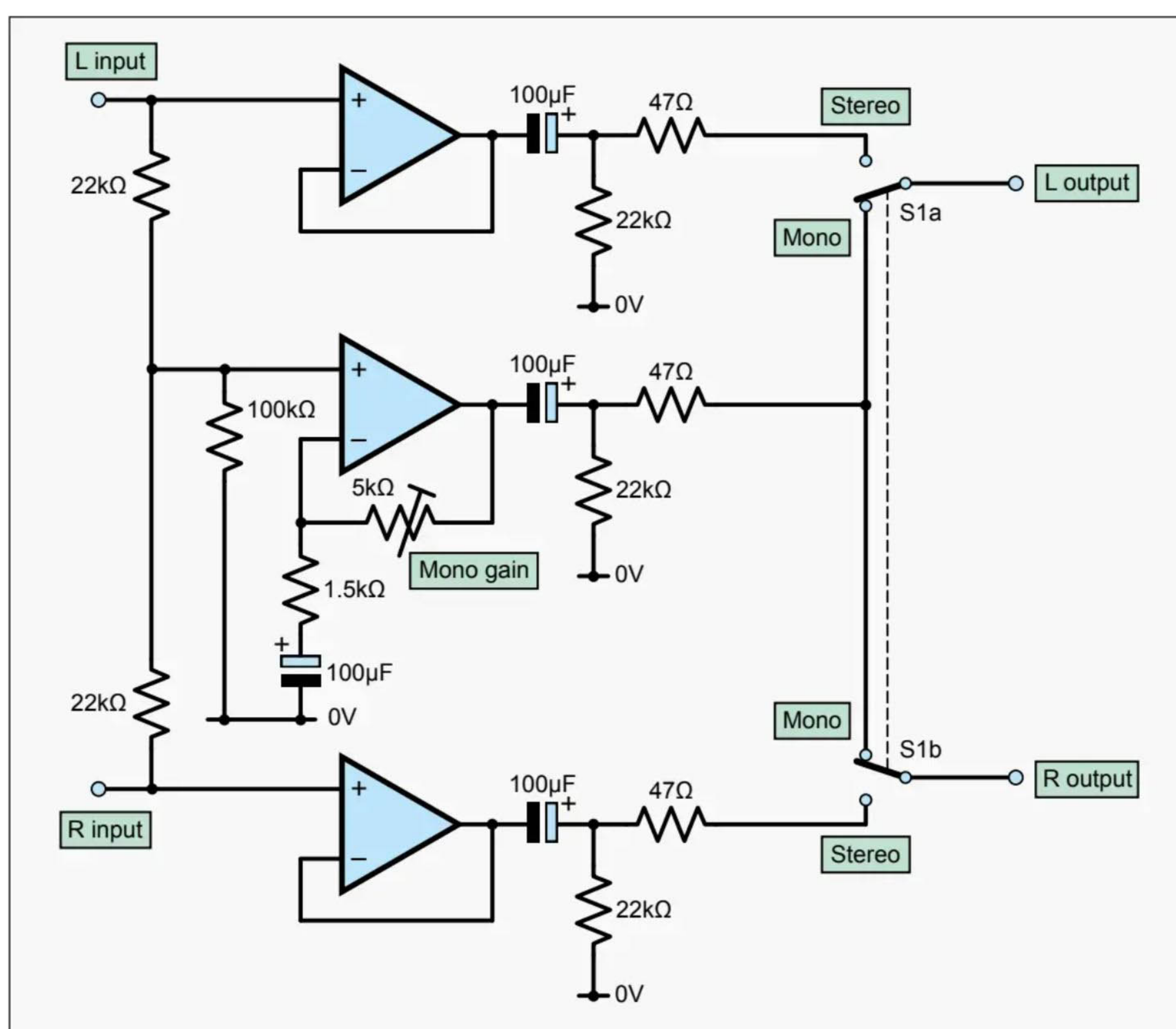


Fig.54. Mono switch circuit with a bit of gain tweaking to equalise the volume difference between mono and stereo.

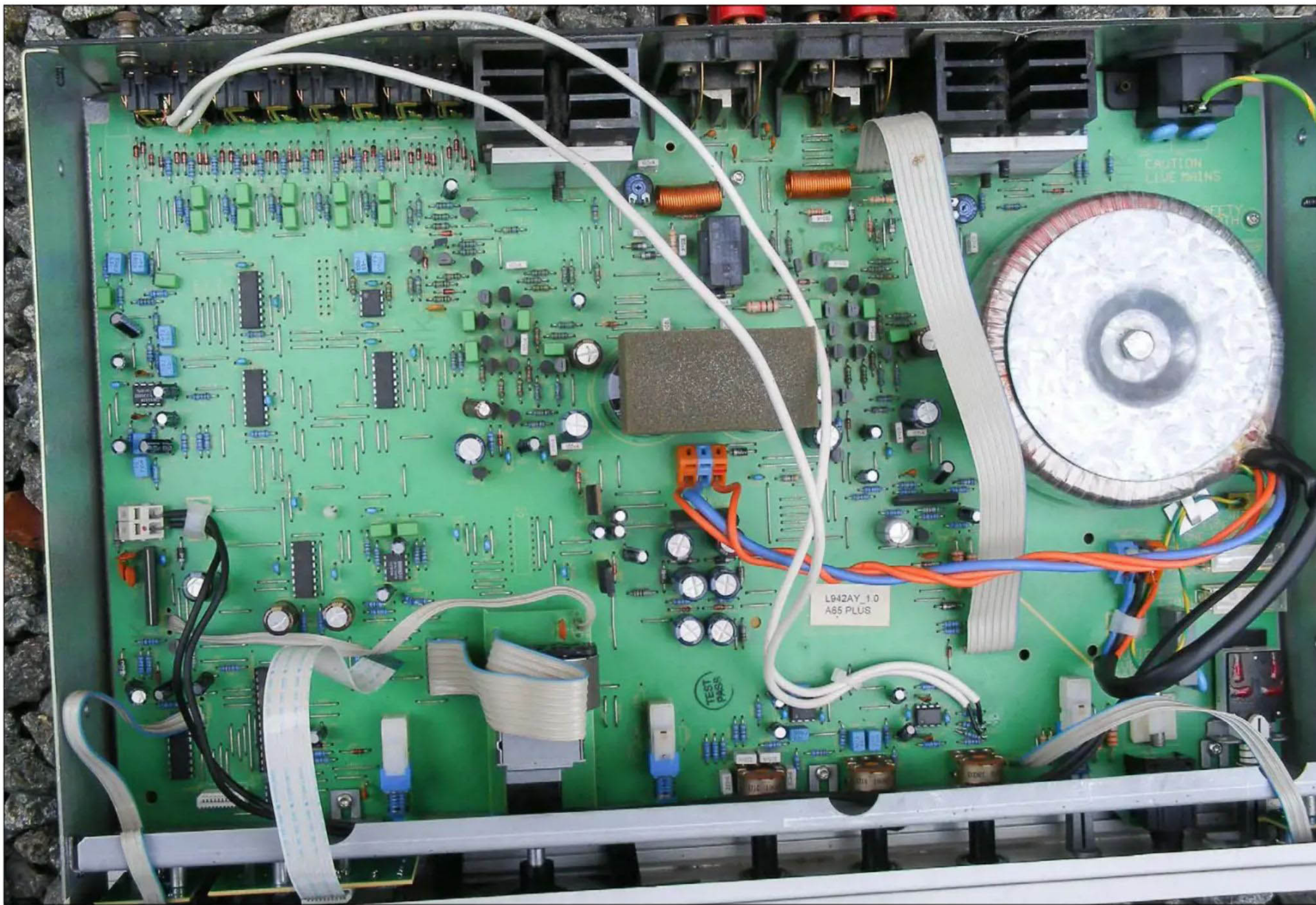


Fig.56. Interior of an Arcam 65 Plus amplifier. It used a microcontroller employing a rotary encoder to accomplish full remote control. Too high tech for me, I couldn't fix it. (Lovely Douglas Self power amplifier though!) The thick white wires are a bypass 'bodge' from the Aux input sockets to the tone control inputs.

do it. One option is quite a convoluted circuit called a 'juggler' using three ganged change-over switches (3PDT) as shown in Fig.55.

Electronic switching

That's it for *mechanical* audio switching; now we can move on to the important topic of *electronic* audio switching, which offers the possibility of remote control that has become an essential marketing tool today. I don't rate it that highly as I'm always losing 'remotes' and I don't think a sedentary lifestyle is so wonderful. If I'm going to select a new source, I see no problem in getting up to operate a switch. A modern amplifier following this tradition is the Arcam 65 Plus shown in Fig.56. This design uses a rotary encoder and a microcontroller to control the electronic switching. Unfortunately, the micro died in mine and could not be replaced thanks to proprietary programming/silicon. I

ended up bypassing the whole selector section, reducing it to a single-input computer amplifier.

A real benefit of electronic switching can be had by removing easily corrupted small-signal wiring connected to mechanical switches at the front panel and confining it to the PCB at the rear. The DC *control* signals can then go anywhere around the front panel for the most ergonomic layout. This means there is no need to worry about hum when placing a switch next to the mains switch.

The simplest and highest quality way of implementing DC control is to use relays, but it is debatable if this is truly 'electronic' – more like 'electromechanical'. On the other hand, I tend to view relay control as electronic because it involves electrical control signals. Again, the relays can be located at the back and only the DC control wires need to go to the front panel switch. Relays are most

commonly available with dual change-over contacts and have to be used in the special circuit shown in Fig.57 to achieve a four-position input selector. A possible problem with this approach is that the signal has to go through more than one contact. If the relays are well-sealed (better still hermetic) with gold-alloy contacts then this is not an issue. Relay control is still the choice in most top tier Hi-Fi systems and

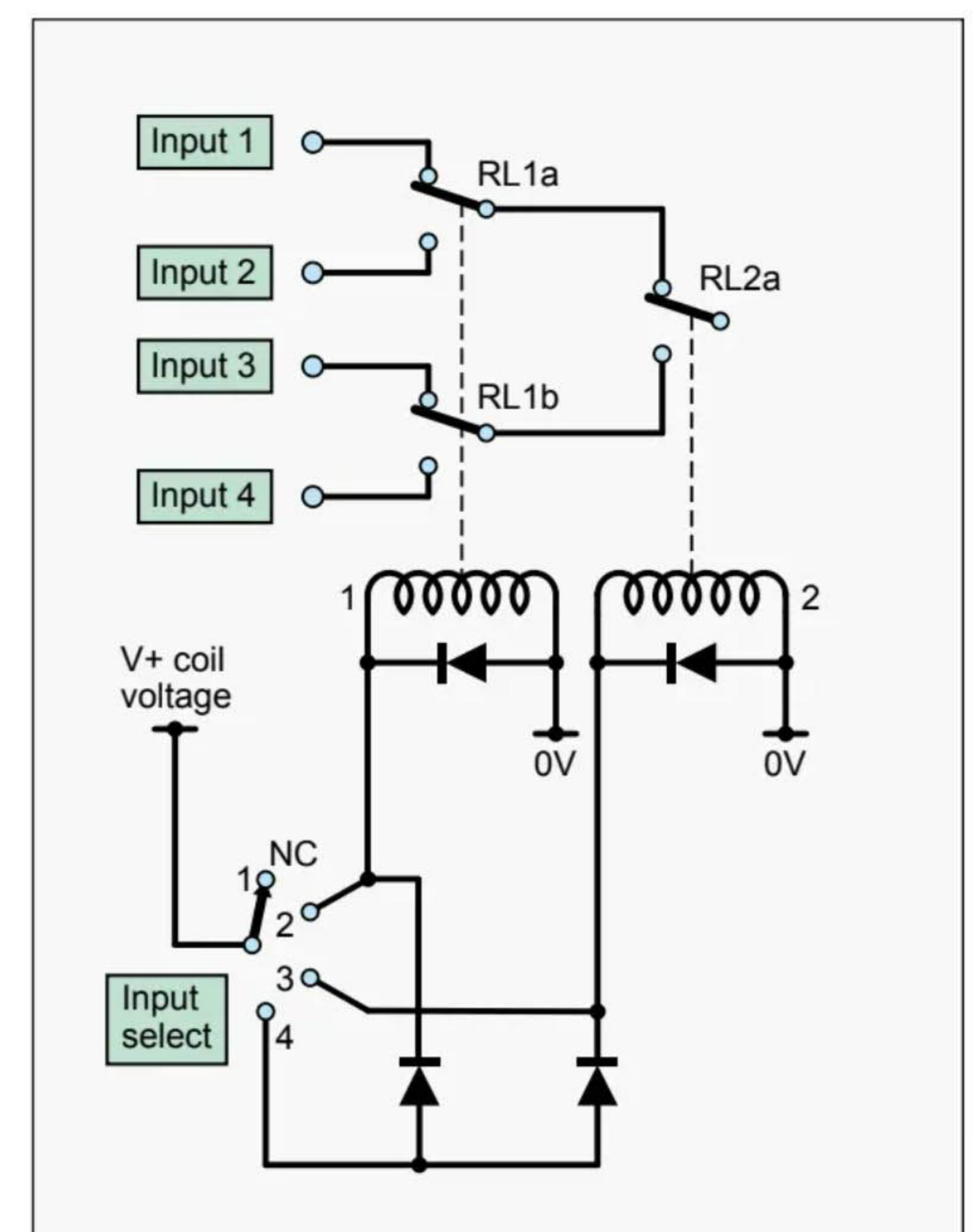


Fig.57. Four input selector switch using DPDT relays. Duplicate for stereo use. Suitable relay is Relpol R8M822N, Rapid 61-6018.

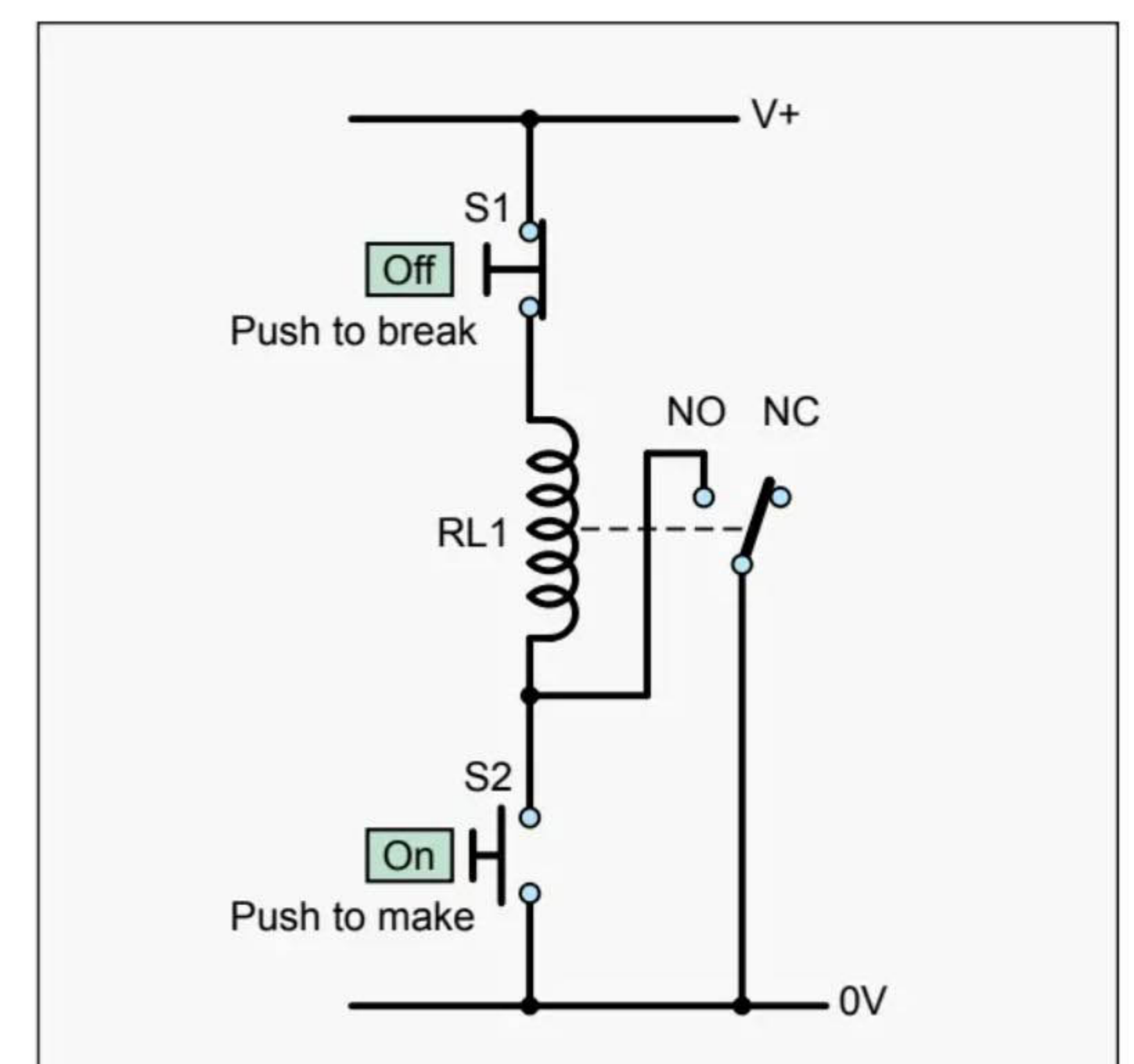


Fig.58. Latching relay circuit.

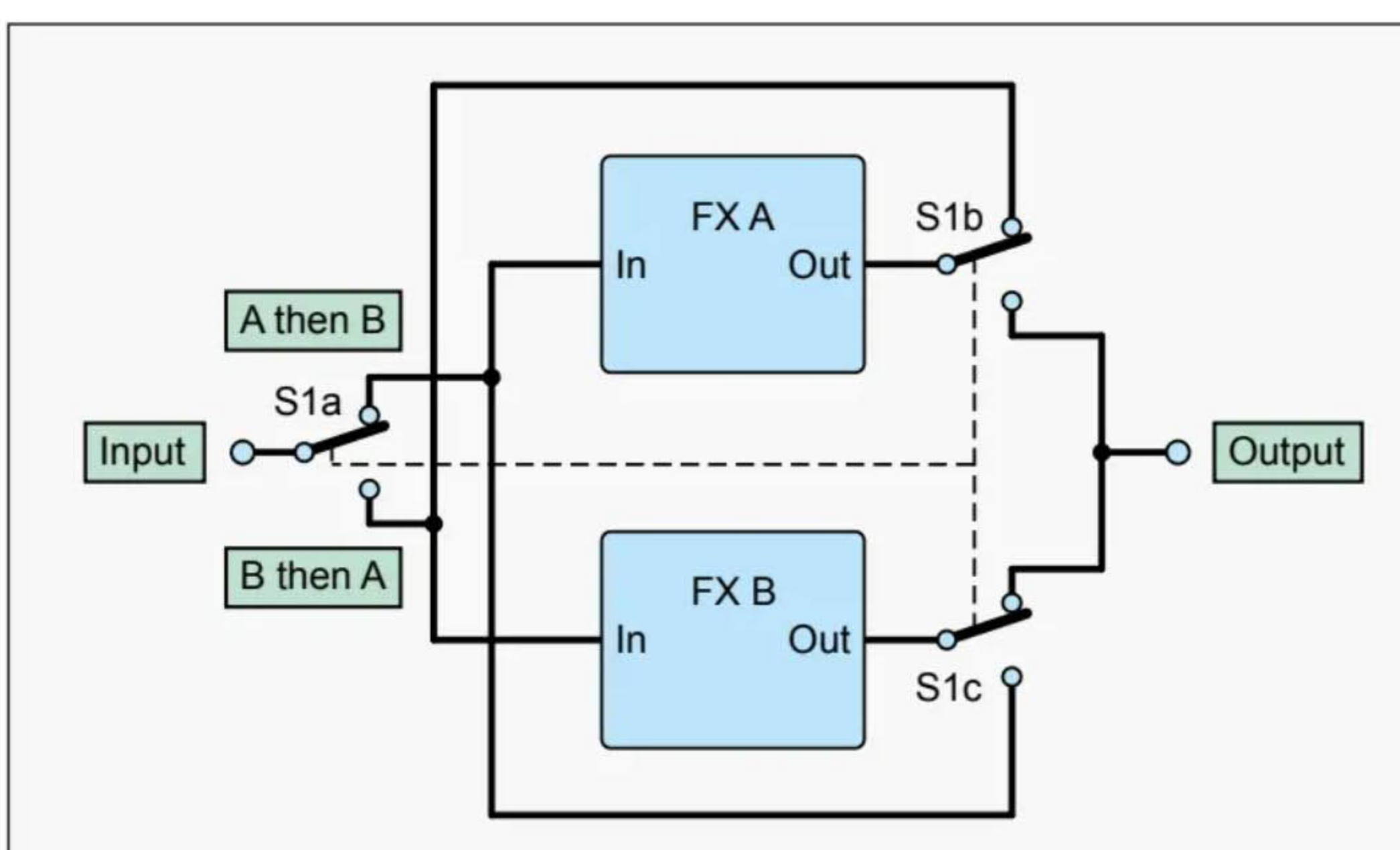


Fig.55. A 'juggler' switch swaps over the order of two circuit blocks. S1 is a 3PDT device. An alternative approach could use a 4053 logic IC or latching relay.

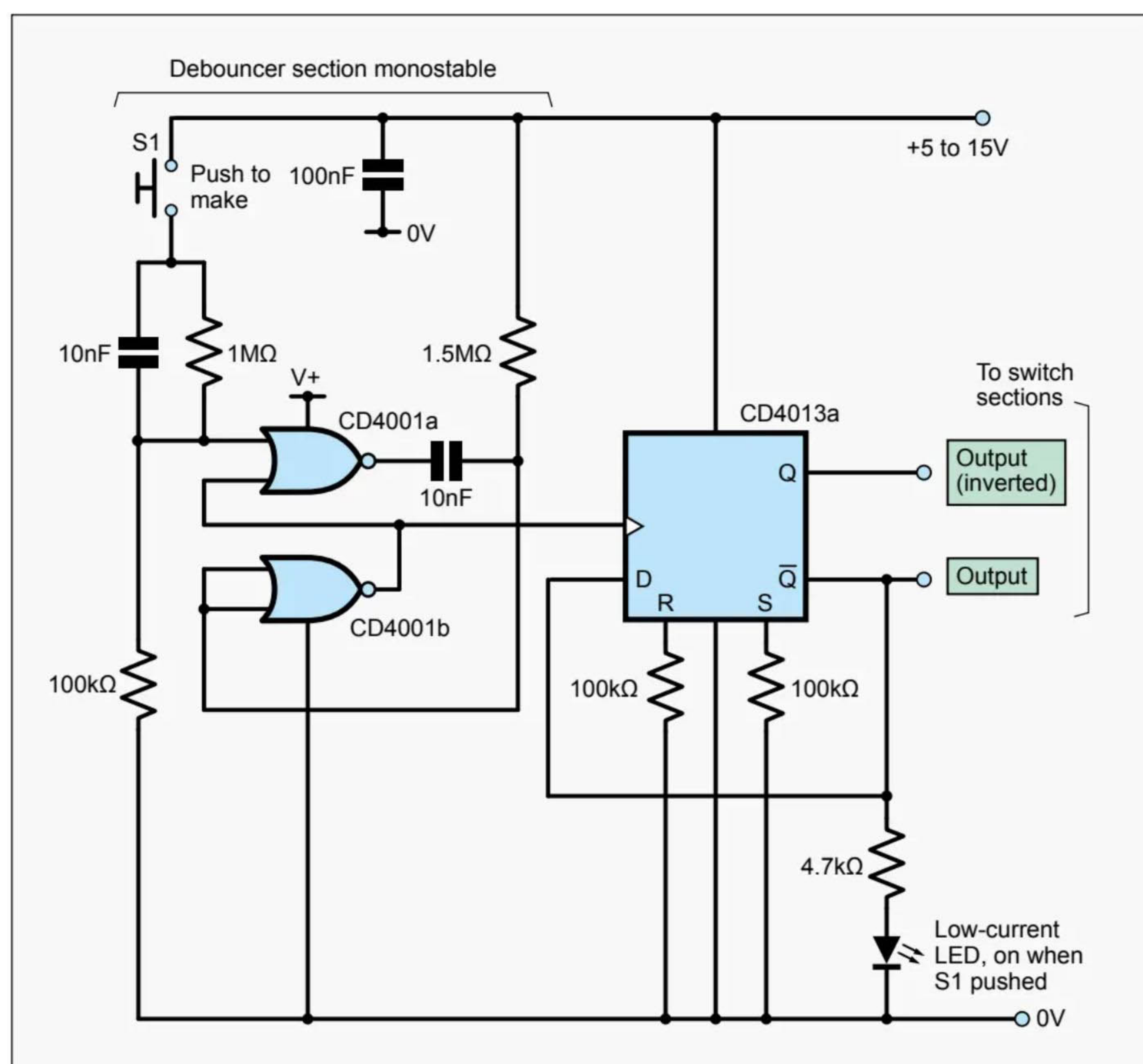


Fig.59. CMOS switch debouncer and flip-flop. Ideal for controlling 4066 chips. Shown using single-rail supply, but could use a dual-rail supply.

test gear. My Audio Precision analyser takes the number of relays to extremes, and many can be heard clicking away as it runs through various test routines. This mechanical noise is annoying and strangely reassuring at the same time. Relay coils consume relatively high levels of power at 100mW or more, and can generate high back voltages when turned off. It is vital to use clamping diodes, an independent power rail and an independent earth to avoid clicks and bangs.

Control logic

One of the great advantages of using electronic control is that latching/maintained switches can be replaced by cheaper momentary types. Gold-plated contacts are not needed either since we are dealing with switching DC rather than low-level signal currents, sometimes called 'dry switching'. However, we need to generate a latching action to hold the selected position. It is possible to use a spare contact on a relay to do this job (Fig.58), but it is more convenient to use a latching logic element, such as a flip-flop/bi-stable to do this. This can be done with standard logic ICs such as the 4013 dual D-type flip-flop. A common problem affecting counting logic when controlled by mechanical switches is switch bounce. This can cause erratic operation, such as jumping a few steps randomly. A monostable made with a couple of NOR gates makes a good de-bouncer to feed the flip-flop, as shown in Fig.59. It's also possible to make a

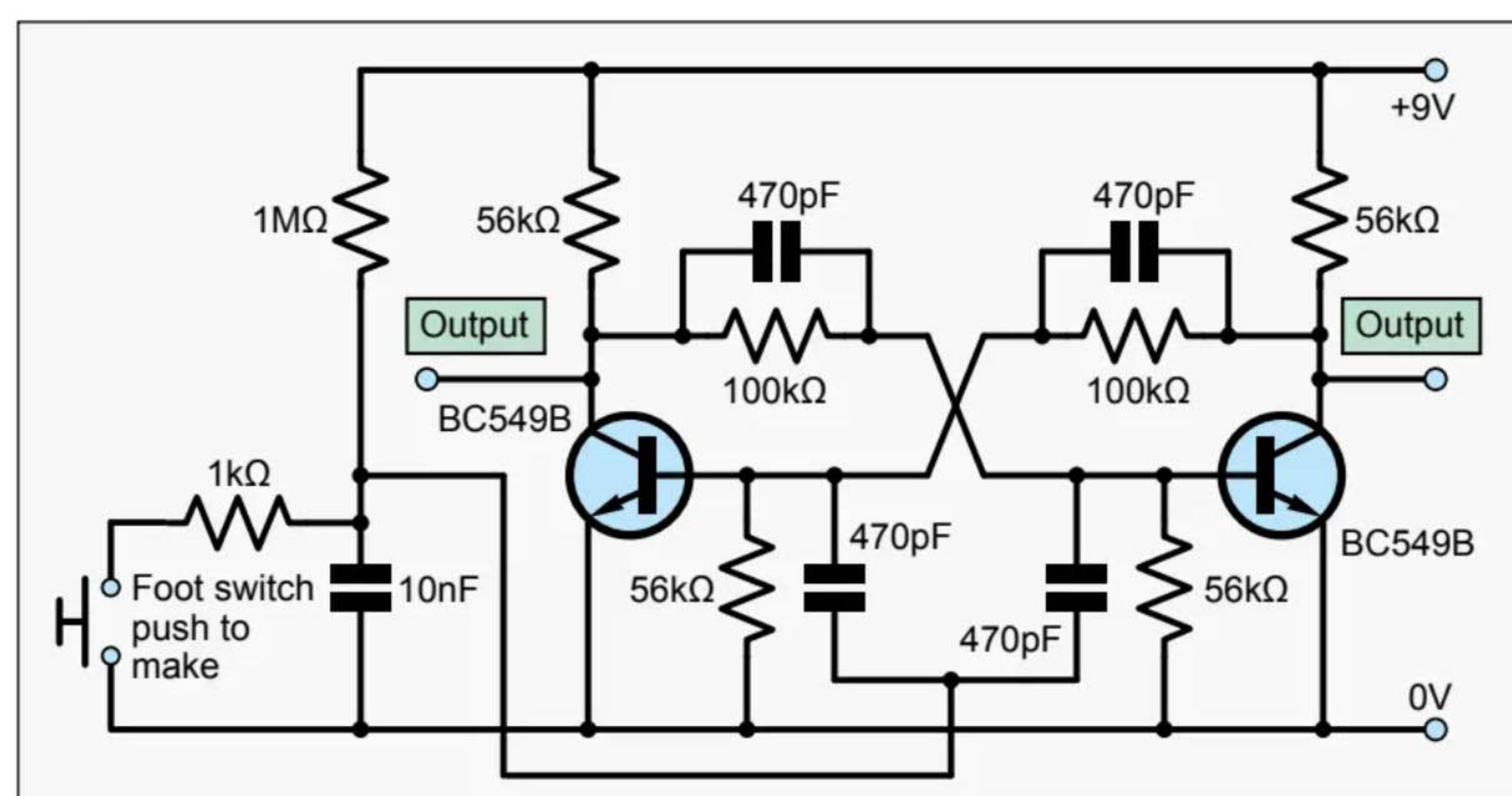


Fig.61. Good old discrete flip-flop; still used in Boss guitar pedals.

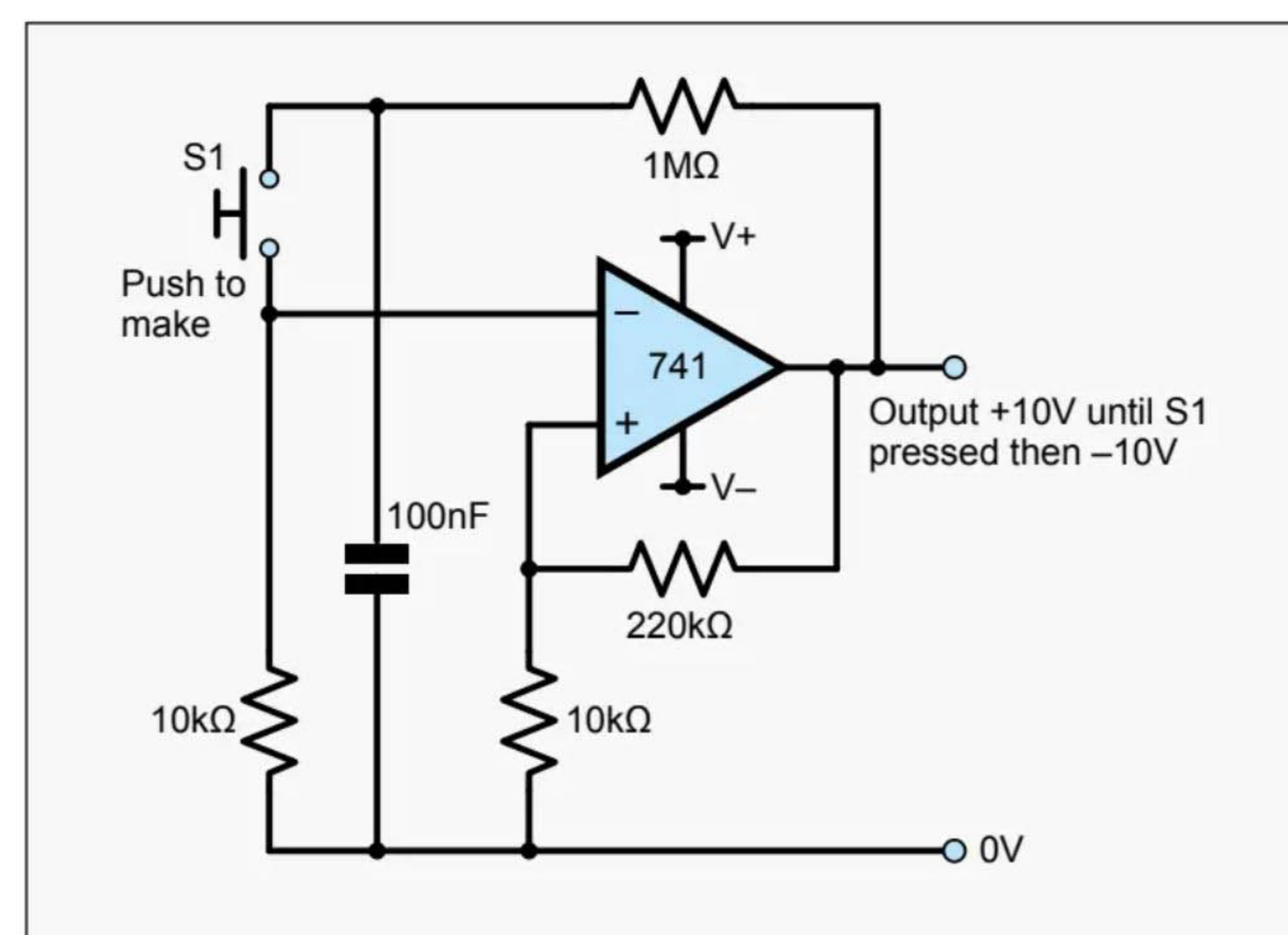


Fig.60. Op amp latching circuit. Avoids the use of unreliable CMOS chips and give higher output voltage as well.

latch with an op amp using a circuit that appeared in *Elektor Summer Circuits* (July 1983) as illustrated in Fig.60. I have found this arrangement useful since it can use a standard op amp power supply. It is also a good use for old spare op amps that are too noisy for audio, such as 741s. A drawback is the problem of an uncertain state upon switch-on. One way round this is to have some kind of capacitive delay on one side of the bi-stable. Fig.61 shows a discrete bi-stable used in Boss effects pedals.

Next month

That was a very brief start to electronic audio switching. Next month, we will dive properly into this important and fascinating topic.

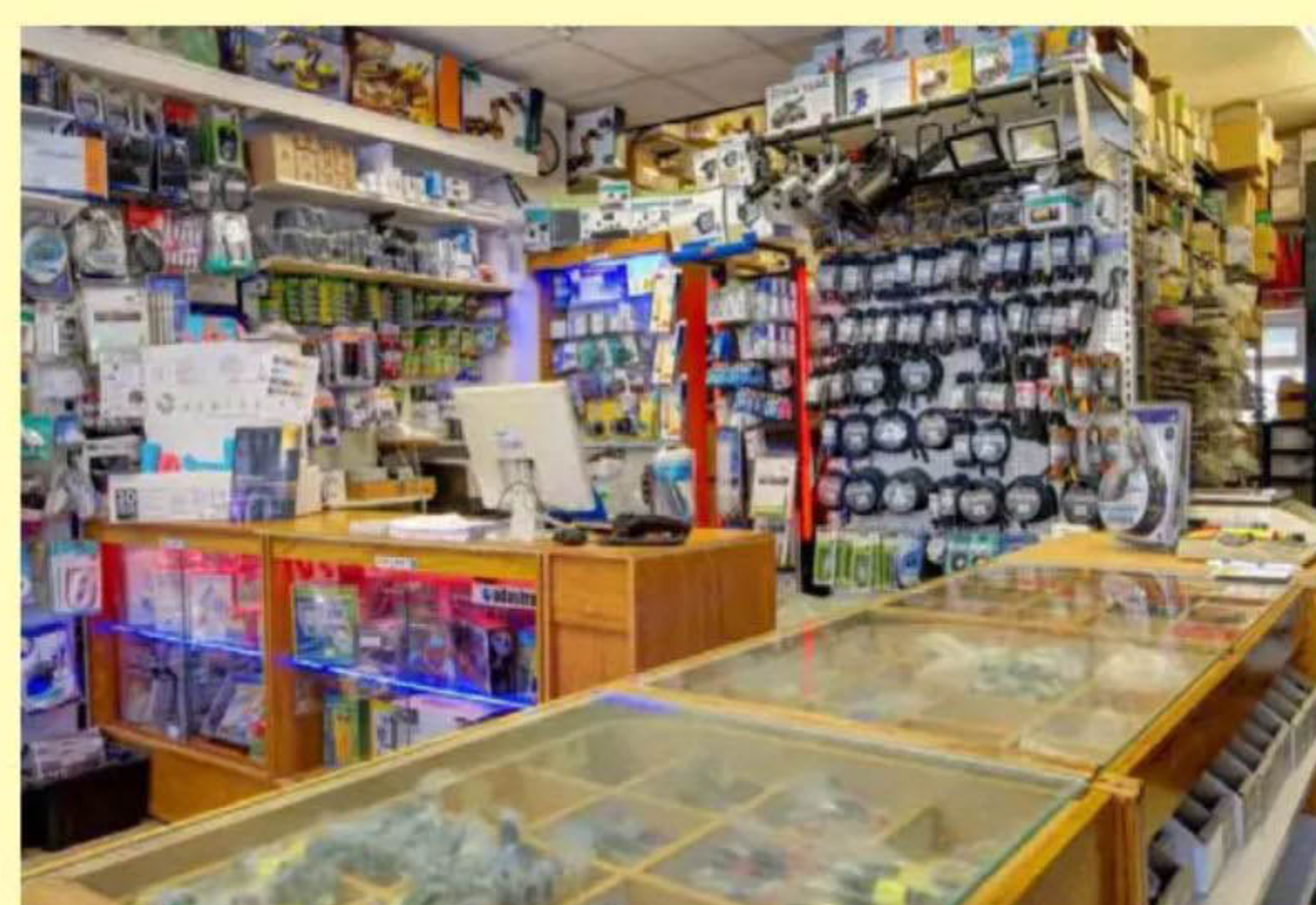
ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.

We can help and advise with your enquiry, from design to construction.

3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more...

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30



**Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ**

Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk

Ge semiconductors

Small-signal PNP transistors
NKT214F, OC57, OC59,
2N1377, 2N525 £1.00

Low-noise PNP transistors
GET106 £2.50

Small power PNP transistors
AC153, AC153K, AC188, AC188K £1.50

Small power NPN transistors
AC176, AC176K, AC187K,
AC141K £2.00

High-voltage PNP transistors
OC77, CV7001 £1.50

TO3 PNP power transistors
OC22, CV7054 (OC23), OC25,
OC35, OC36, AD143, AD149,
AD161, AD162 £2.00
AD140 £3.50
AD149, AD161/2 matched pair £5.00

RF PNP transistors
OC41, OC42, 2SA12, 2SA53, AF124,
AF178, GET872A £1.50

NPN transistors
OC139, OC140, ASY73 £2.00

Diodes
CV7049 (OA10), CG92 (OA91) £0.50

Si semiconductors

Diodes
ZC5800 RF Schottky £0.20

Low-noise silicon transistors
BFW16A, 2SC3071, 2SC3068,
2SA1016K, 2SC2362K,
2SA970BL £1.50
2SC2204, 2SD655, BC550C £0.50
ZTX651 £0.30
ZTX751 £0.50

RF transistors (suitable for Theremin)
BF199 £0.50

Audio power MOSFETs
Exicon 10N20, 10P20 £6.50
Hitachi 2SJ99, 2SK343 £3.50
Hitachi 2SJ56, 2SK176 £8.50

JFETs
BFW11, BFW10, TIS73L, J177,
J113, U1994, U1898, 2SJ176,
J201 £1.00
J175, J176, J112, J111, 2N3820,
2N5467, BF244, 2N5460, J230 £0.60

Small power output/driver transistors
2SB649A, D669A, 2SA1208,
2SC2910, MJE253G, MJE243G,
2SA1725, 2SC4511 £1.20
BD139, BD140, BD135, BD156, BD435,
BD436, MPSA63, BCV46 £0.50

MOSFETs
ZVP2106A £0.30
Dual-gate 3SK45, BFS28 £1.50

Metal-cased transistors
BC143 £0.35
2N1711 £0.50
BCY71 £0.30
BC109C £0.60

Amplifiers
LM318 high-speed op amp £0.35
µA709 metal-cased op amp £2.00

5534H metal-cased op amp £3.00
LM384, TDA2030A,
TDA2050V £1.50
TAA435 (Mullard power
amp driver) £1.20

Synthesiser ICs

That/dbx 2180 VCA/VCF £6.00
CA3080 VCA/VCF (vocoder) £2.80
CA3280 VCA/VCF £4.00
LM13600/ LM13700 VCA/VCF £2.00
CA3086 transistor array £1.00
PT2399 echo/delay £1.50

Dual transistors

2N2639, 2N2223, 2N2910 (NPN) £4.00
E401 (JFET Moog) £4.00
2SK2145-Y dual JFET £0.80
2N5564 JFET £8.00
HN3C51F, HN3A51F £1.00
DMMT3904/6, HN1A01F,
HN1C01F £0.50

Loudspeakers

PE Mini-Monitor
Volt PE165 6.5-inch woofer (each) £85
Morel MDT29 tweeter (each) £25
Kit pair of PE165/MDT29 plus
Volt crossover parts and PCBs £299
Monacor DT-28N tweeter (each) £35
Vifa 19mm BC20SC15-04
tweeter (each) £15

Volt crossover inductors
1.2mH, 1.5mH, 2mH, 2.7mH, 0.5mH
(tapped at 0.3mH) (each) £5.85

*Fully assembled and tested high-quality
speaker prototypes – ask for details*
LS3/5As and other similar speaker
systems (pair) £200-£350

Fully tested reclaimed speakers
Vifa BC14 5-inch woofer (each) £10
Vifa TC26 1-inch tweeter (each) £10

Low-price speakers
Philips 4-inch 4070 £2.00
EMI 10x6-inch, 30Ω Alnico £7
64mm 64Ω neodymium £1.20
5x3-inch elliptical 50Ω or 80Ω
Alnico £3.50
1.65x2.75-inch 8Ω £1.50

Capacitors

Note '10/63' denotes '10µF 63V'.

Polyester
3.3/100, 4.7/250, 4.7/63 £1.00
5.6/63, 8.2/63, 10/63 £2.00
Mullard 'Mustard' C296 0.22/400 £2.00

Polycarbonate
Axial 2.2/63 1%, 4.7/160, 6.8/63 £1.00

Radial
6.8/160V, 10/63 £2.00
Reclaimed 22/63 £2.00

Polystyrene
Philips 1% 4.7nF/160, 6.2nF/500,
12nF/63, 22nF/63, 110nF/63,
24nF, 2nF £1.00
RIFA 1% 100nF/100, Suflex
90.9nF 0.5% £2.00
Suflex 2.5% 10nF/63 (rad. or ax.) £0.50

Radiation resistant
Siemens cellulose acetate MKL
2.2/25 £0.80

Electrolytic – Mullard blue 017 series
10/25, 22/25, 100/10 £0.50
150/40, 470/40, 1000/40 £1.00

Tantalum – axial metal cased
22/50, 47/35, 68/25, 100/20, 120/10,
150/16, 220/10, 330/6 £2.00
22/35, 33/35, 47/20, 68/15, 100/10,
150/6 £1.25
4.7/50, 6.8/35, 10/25, 10/35, 22/15 £1.00

Axial moulded-case tantalum
Kemet axial 6.8/10 £0.30
Kemet radial 33/10 £0.40
STC radial 100/20 £1.50

Tantalum bead
22/50 £1.00
470/3 £2.00
680/6.3 £3.00

Wet tantalum
220/25 axial £3.20
Castanet button 140/30, 470/3 £3.20
Hughes 540/10 £3.20

Bipolar
Hermetic bipolar tantalum 16/35 £3.20
Elcap axial 10/50 £0.50
Generic radial 100/16, 470/35,
100/ 63, 22/35, 4.7/35, 220/16 £0.50

Philips solid-aluminium (axial) 121/123
47/16 £1.00
330/6.3 £2.00
100/35 £4.00

Philips Pearl 122 series (radial)
10/16 £0.30

Silvered mica (radial)
1nF/500 1% £1.00

Trimmer capacitors
Vishay plastic-film 4-40pF £0.80
Vishay plastic-film 5.5-45pF £1.00
Vishay plastic-film 5-80pF £1.20
Vishay plastic-film 10-250pF £2.00
Mica 1-12pF, 2-40pF £1.00

Audio transformers and inductors

Eagle transformers
LT44, LT722 driver, LT700, LT723
500Ω output £2.50
LT30 500mW output £3.50
5:1 interstage £1.50

Repanco
T/T3 splitter transformer £4.00
CH2 5mH RFC £2.00

Balanced output transformer
Vigortronix 600Ω VTX-101-007 £10
Vigortronix 600Ω VTX-101-3001 £10
Vigortronix 600Ω VTX-101-3002 £15
Gardners 150Ω £10
Reclaimed BBC LL74/MPC
nickel core 600Ω £12

Reclaimed mic input transformer £15

Inductors
82µH, 4.7mH, 100µH, 270µH,
10µH, 14µH (low Z) £0.50
7-inch ferrite rod with MW
and LW windings £3.00

Special resistors

Bourns wire-wound trimmer
5kΩ 3059 JM panel-mount £2.00
Thermistor RA53, R13 £4.00
A13 £2.00
Thermistor CZ1, CZ6 £1.50
Holco H2 2.2MΩ 1W, 1% £1.00
Welwyn 1GΩ 2W £1.00
5k Bourns 3321H cermet trimmer £0.50

Potentiometers

Bourns 81
25kΩ lin cermet £2.00
25kΩ lin, 5kΩ lin conductive
plastic £3.00

Bourns 91
10k dual-gang lin or log £10.00

Plessey moulded-track
5kΩ log with switch £3.00
50kΩ A/log £3.00
Mil 250kΩ lin dual £5.00

Alpha
16mm 4.7kΩ A/log £0.80
220kΩ A/log £0.80
10kΩ lin centre-detent dual-gang £1.50

Allen Bradley J series/Honeywell
10kΩ lin 1 million cycles £5.00
Blore Edwards AB 45 dual 5kΩ
A/log with switch £3.50
Alps RK9 dual-gang 5kΩ RD law £4.00
Alps 50k log tapped motorised
stereo £5.00
BI P260 500kΩ log conductive
plastic 1 million cycles £2.00
Colvern wire-wound 100kΩ or
50kΩ dual-gang 3W £5.00
Mil Spec hermetic 10Ω £8.00

Miscellaneous

Theremin Clearance Sale!
Elysian Theremin MIDI box £300

PCBs
Pocket Theremin (EPE, 1996) £2.00
Elysian Theremin (EPE, 1996) £6
Synth VCF, VCO (EPE, 2017) £3.00
48V PSU (EPE, 2019) £3.00
SMT dual transistor adapter £0.40

Contact

Jake Rothman
The Old Rectory, Arlais Road,
Llandrindod Wells, Powys
LD1 5HE (visit by appointment)
+44 (0)1597 829102
jrothman1962@gmail.com

Minimum order £5.00 inc post
Quantity discounts negotiable

Payment

PayPal, cards (via phone), bank
transfer, cheques (payable to 'J
Rothman', UK pounds only)
No VAT payable

Postage

Small Jiffy bag £2.99
Small package £4.99
Big boxes and overseas at cost
– ask for a quote



Arduino Bootcamp – Part 20

Just for giggles and grins, let's start this column with what they call a 'pop quiz' in the US (this means an unexpected test; it's nothing to do with pop music, thank goodness). The way we currently have our clock working, we always display the most-significant (left-hand) digit, even if it's a 0 (Fig.1a). Suppose we decided to suppress this leading 0 and instead display a blank (Fig.1b). Can you think of a way to implement this in our latest and greatest program from our previous column (*PE*, July 2024) using the minimum number of modifications?

Dazed and confused?

With respect to the 7404 and 7408 integrated circuits (ICs) we introduced in our previous column, I received several emails from dazed and confused readers asking about the extra letters marked on the packages, which should read something like SN74LS04N and SN74LS08N. For one stalwart fellow, it was the 'N' that threw him into a tizwoz. 'Have I ended up with the wrong ones?', he pleaded, plaintively.

Let's start with the first two letters, which are shown as 'SN' here. These refer to the manufacturer, which is Texas Instruments (TI) in this case – under TI's part numbering scheme, 'SN' stands for 'Semiconductor Network'. The '74' portion of the moniker indicates commercial components that operate over a temperature range of 0°C to +70°C. If this were 64, it would indicate an industrial component (–40°C to +85°C), while 54 would signify a military component (–55°C to +125°C).

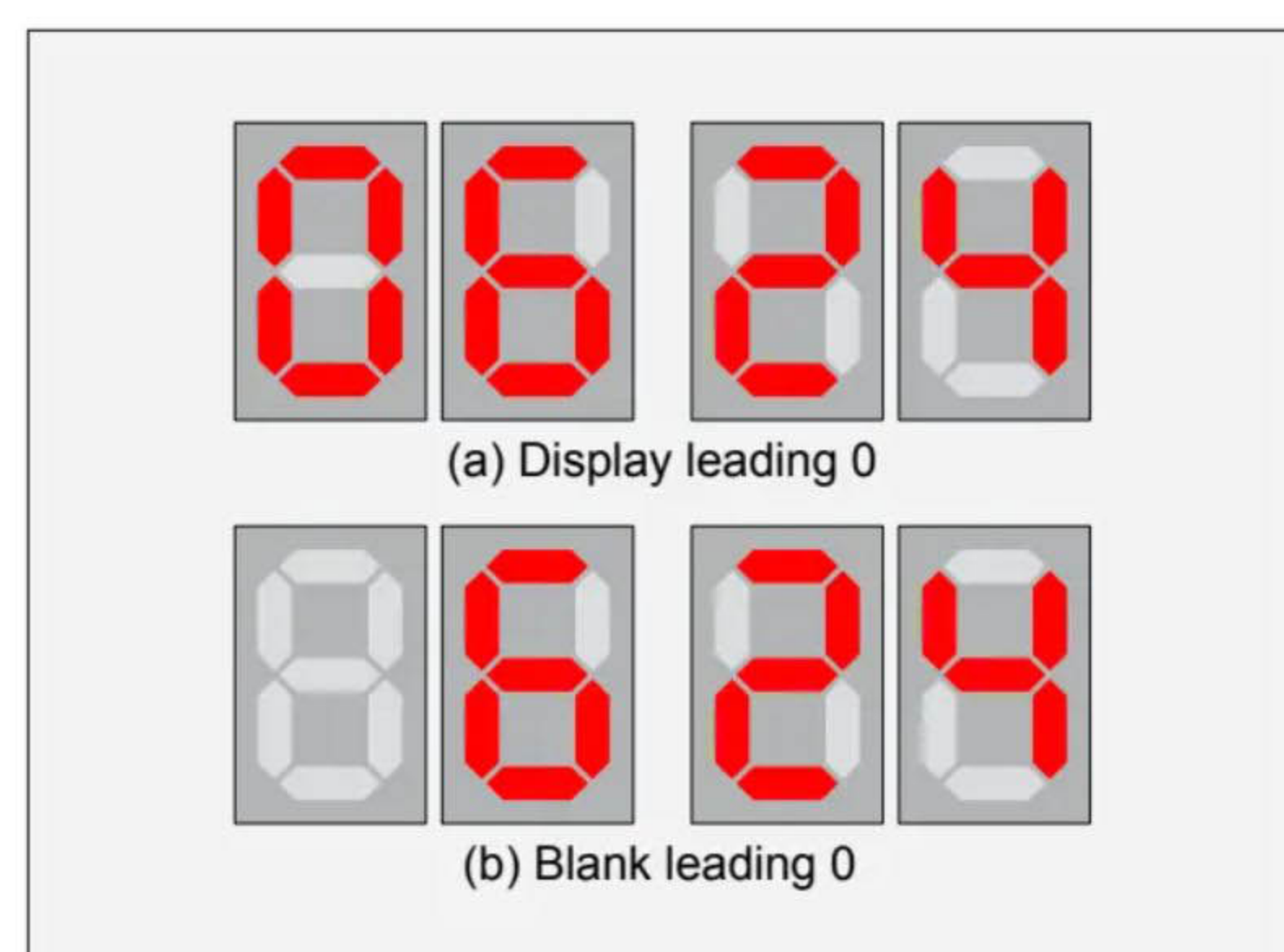


Fig.1. Do we want a leading 0?

There are many different families of these components. I suggested the LS ('Low-Power Schottky') family because they are cheap and cheerful and do the job. My second choice would have been the HC ('High-Speed CMOS') family because these are more tolerant with respect to the voltages on their input signals. On the other hand, they are less tolerant of static, and they may be a tad more expensive. Some hobbyist kits offer both LS and HC types – see: <https://bit.ly/44JMZRa>

The 04 and 08 reflect the types of logic functions (the 04 contains six NOT gates; the 08 boasts four 2-input AND gates). You can peruse and ponder the entire list of available functions on the interweb at: <https://bit.ly/3wON1L3>

Last, but certainly not least, the final character (or characters) indicates the package type, where N tells us that our devices are presented in plastic dual in-line packages (PDIPs).

If you wish to delve deeper, a handy-dandy *Interpreting TI Logic Part Numbers* document is available for your perusing pleasure at: <https://bit.ly/3ylkYn3>

Little and large

There used to be a British comedy duo called Little and Large. Straight man Syd Little (born Cyril John Mead) and comic Eddie Large (born Edward Hugh McGinnis) became household names in the late 1970s and the 1980s. The reason they just popped into my mind is that we tend to deal with very small and very large values with respect to electronic components (just call me the 'Sovereign of Segues').

Take resistors, for example. The values (resistances) of these little scamps are measured in units of ohms (symbol Ω). These are named after the German physicist Georg Simon Ohm, who studied the relationship between voltage, current and resistance (and often forgotten, temperature). We typically deal with resistors having large values measured in hundreds of ohms, thousands of

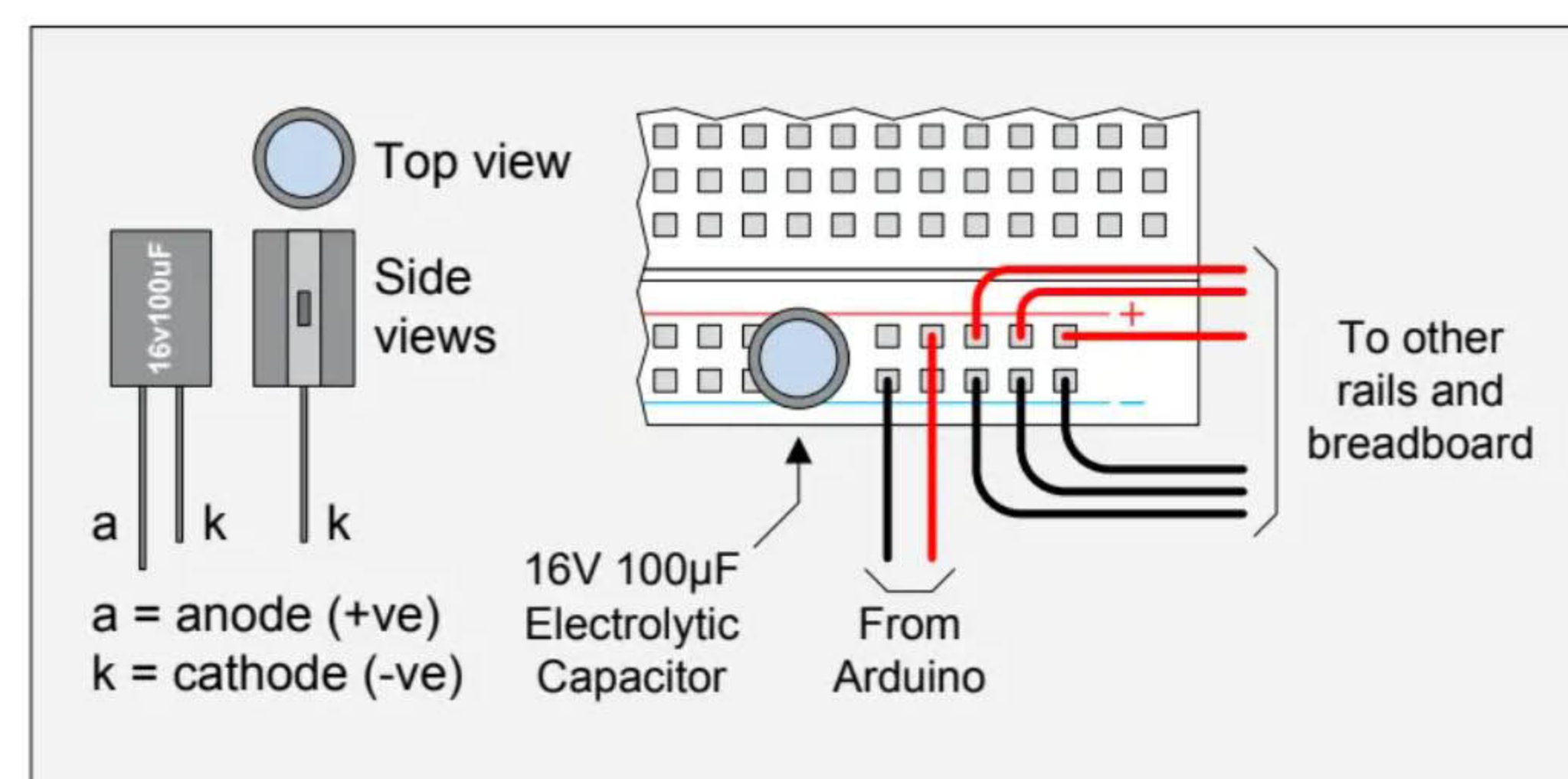


Fig.2. Adding a 16V 100µF electrolytic capacitor.

ohms (kiloohms or $k\Omega$, meaning 10^3), or millions of ohms (megaohms or $M\Omega$, meaning 10^6).

By comparison, in the case of capacitors, their values (capacitances) are measured in units of farads (F). The term 'farad' was originally coined by Latimer Clark and Charles Bright in honour of the English scientist Michael Faraday. We typically deal with capacitors having small values measured in millifarads (mF, meaning 10^{-3}), microfarads (μF , meaning 10^{-6}), nanofarads (nF, meaning 10^{-9}), and picofarads (pF, meaning 10^{-12}).

Why am I waffling on about this now? Well...

Smoothing things out

As I mentioned at the end of our last column, I ran into a WTW ('what the what?') problem with one of my breadboards. During the process of diagnosing this problem, I did what I should have done much earlier, which is to add a 16V 100µF electrolytic capacitor straddling the power and ground rails on the breadboard as close as possible to the point where the power and ground wires arrive from the Arduino (Fig.2).

This type of capacitor looks like a small soda can with two legs. It's a polarised component (ie, a part with polarity), which means it can be correctly connected into the circuit in only one orientation. The longer lead is the anode, which is connected to the more positive rail (the +5V rail, in our case). The shorter lead is the cathode, which is connected to the more negative rail (the 0V rail, in our case). The cathode side of the can is also marked with a minus sign to give us a clue. You can obtain these devices

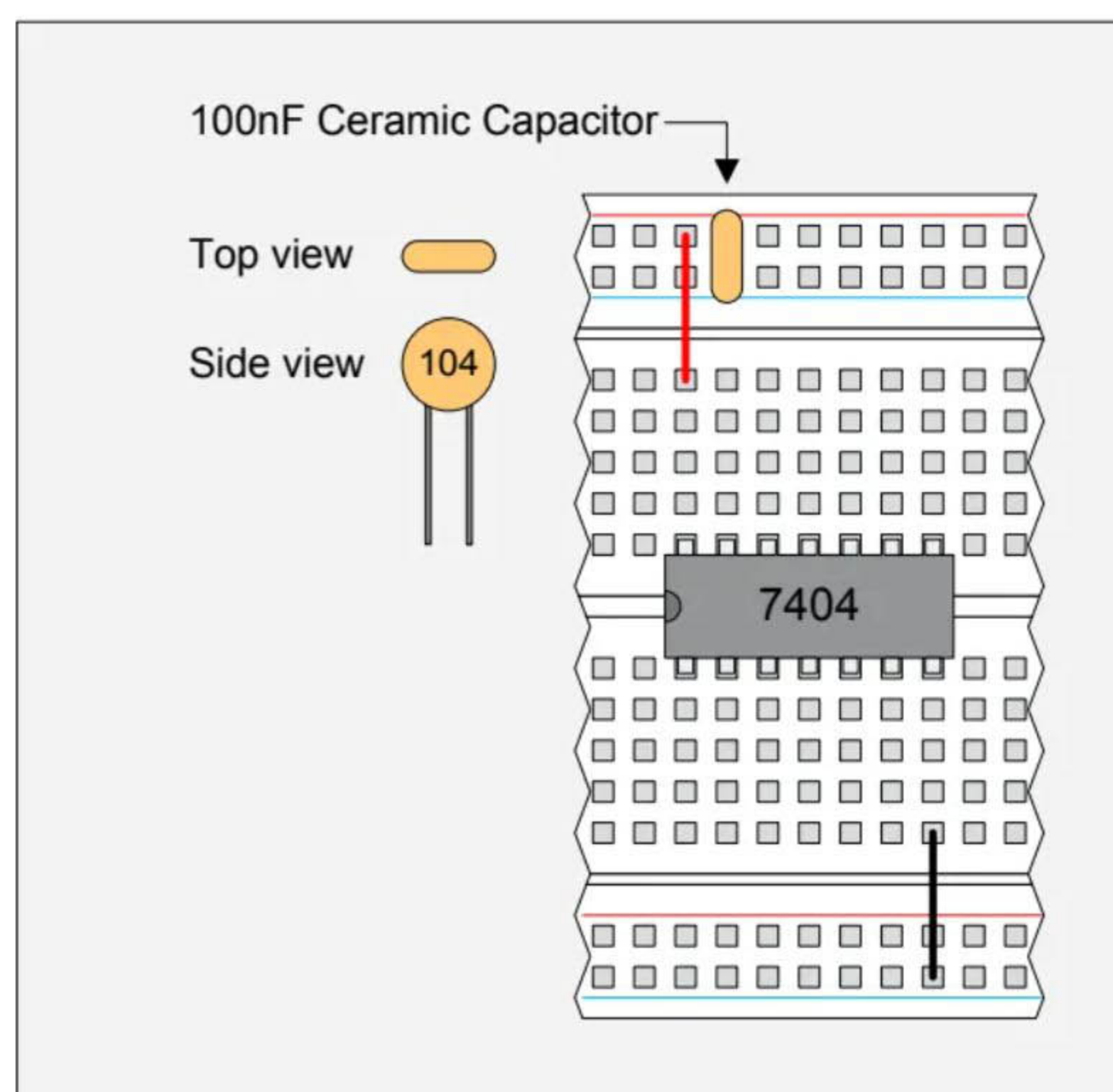


Fig.3. Adding an 0.1 μ F (100nF) ceramic capacitor to the breadboard.

from any electronic component supplier, including Amazon – see, for example: <https://bit.ly/44LzpNa>

The 100 μ F is the device's capacitive value read as '100 microfarads'. The 16V is its rated value, which means it isn't capable of handling more than 16V. I picked 16V because (a) it provides nice headroom over our 5V supply and (b) it's a commonly available value. It would be perfectly OK to use a higher rated component like 25V or 50V if that's what you have available. However, a higher voltage rating typically means a physically larger or more expensive part, often both.

This capacitor is used to filter out any electrical noise from the power source, which is the Arduino in our case. It can also help to smooth out any power dips caused by components switching and drawing extra current. This isn't a mission-critical component for our existing circuit, but I've decided to add one anyway.

No coupling allowed

As you may recall, the terms 'current-limiting,' 'pull-up,' and 'pull-down' are simply qualifiers we add to resistors to remind ourselves as to the functions we're using them to perform (the physical devices we use are the same). Similarly, ceramic capacitors can be employed for a wide variety of tasks, including acting in 'bypass' and 'decoupling' roles. Although these serve different purposes, many people use the terms 'bypass capacitor' and 'decoupling capacitor' interchangeably.

A 'bypass capacitor' is usually applied between the power and ground pins of an IC. It provides a low-impedance path to steer voltage spikes and high-frequency noise to ground, thereby 'bypassing' the IC. (This noise can come from the power supply or other parts of the circuit or external sources).

The term 'coupling' refers to the undesired transfer of electrical energy between subsystems. For example, when

digital logic devices – like our 7404 and 7408 chips – are switching, they can pull the supply rail voltage down by introducing a momentary (transient) current load. This can affect nearby devices, which are 'coupled' via their power and ground pins. To prevent this, a 'decoupling capacitor' is usually applied between the power and ground pins of the IC (the same as the bypass capacitor). In this case, we can think of the capacitor as acting like a small, localised energy reservoir (a miniature battery, if you will), holding the voltage steady long enough for the main supply to catch up.

It's a good idea to add decoupling/bypass capacitors in the form of ceramic components to our circuits. If you are unsure as to the number(s) and value(s) of these capacitors, you should check the data sheets for the specific manufacturers' parts you are using. Having said this, a good rule of thumb is one 0.1 μ F (100nF) capacitor per IC. This should be positioned as close to the IC's power pin(s) as possible, connecting this pin to the ground plane in the case of a printed circuit board (PCB), or to the ground rail in the case of a breadboard (Fig.3). Ceramic capacitors aren't polarised, so we can connect them either way round.

Why haven't I mentioned these before? Well, our current circuit involves only two chips containing combinatorial functions that are less susceptible to noise and voltage dips on the power rail. Things will become more 'interesting' when we start to use chips containing sequential (clocked) elements, such as shift registers, for example. Having said this, it's good to get into the habit of adding these capacitors as a matter of course, so that's what we are going to do. You can get ceramic capacitor kits containing multiple values – see: <https://bit.ly/4bEAUiv>

One thing that can trip beginners up is that the values printed on these components are not particularly intuitive, so do your best to not get them mixed up. What we are looking for is components that say '104' (if we are lucky) as illustrated in Fig.3. What does this mean? I was hoping you wouldn't ask. It means 10 x 10 to the fourth power (10×10^4) = 100,000 measured in picofarads (pF), which equates to 100nF or 0.1 μ F (now, *you* try explaining this to someone).

I've added all the capacitors mentioned here into the latest and greatest incarnation drawing of our prototyping platform (file **CB-Aug24-01.pdf**). And, as usual, all the files mentioned in this column are available from the August 2024 page of the *PE* website: <https://bit.ly/pe-downloads>

Mea culpa

I have something to share with you – *mea culpa* – which means 'through my fault' in Latin, and is used to mean 'it was my fault' or 'I apologise.' So, why am I apologising? Well, I came to realise that I sort of threw you in at the 'deep end' in our previous column. I presented you with the schematic and layout for our new 2:4 decoder without any accompanying assembly and test instructions... yup, *mea culpa*

Throughout this series, I've enjoined you to take things one step at a time. I should have followed my own advice when it came to the addition of the decoder. 'It's just two jellybean logic chips,' I thought. 'I can do this in my sleep,' I thought. 'What could possibly go wrong?' I thought.

So, you can only imagine my surprise when I first ran the program with the circuit featuring our new decoder and... it didn't work. Even worse, it didn't work in a very unintuitive way. That's when I started to worry about how you were getting on at your end.

After a lot of messing around trying to diagnose the issue, eventually resorting to having a video call with my chum, Joe Farr, it turned out there was a problem with the internals of my breadboard in the area where I'd added my 7404 and 7408 chips (sometimes one is tempted to wonder if Gremlins truly are naught but a myth).

When I eventually come to remove all the components from this breadboard at some stage in the future, I'm going to pry the back off and look inside to locate the problem. However, since I was pushed for time, and since I knew that everything *apart* from the decoder was operating as expected, I simply reconstructed the decoder portion of the circuit on a separate half-size breadboard (Fig.4).

As you can imagine, I was just a little less cocky this second time around, causing me to implement the decoder in a more staged way, as discussed later in this column.

That's shocking!

Before we plunge into the fray with gusto and abandon (and aplomb, of course), we should remind ourselves that semiconductor devices like light-emitting diodes (LEDs), transistors and ICs can be fatally affected by static electricity in the form of electrostatic discharge (ESD). If you happen to build up a static charge (by walking across a carpet in your stocking feet, for example) and you generate a spark, although there's not much current, that spark can be as high as 35,000V, which can give any semiconductor device with which it comes into contact a very 'bad hair' day indeed.

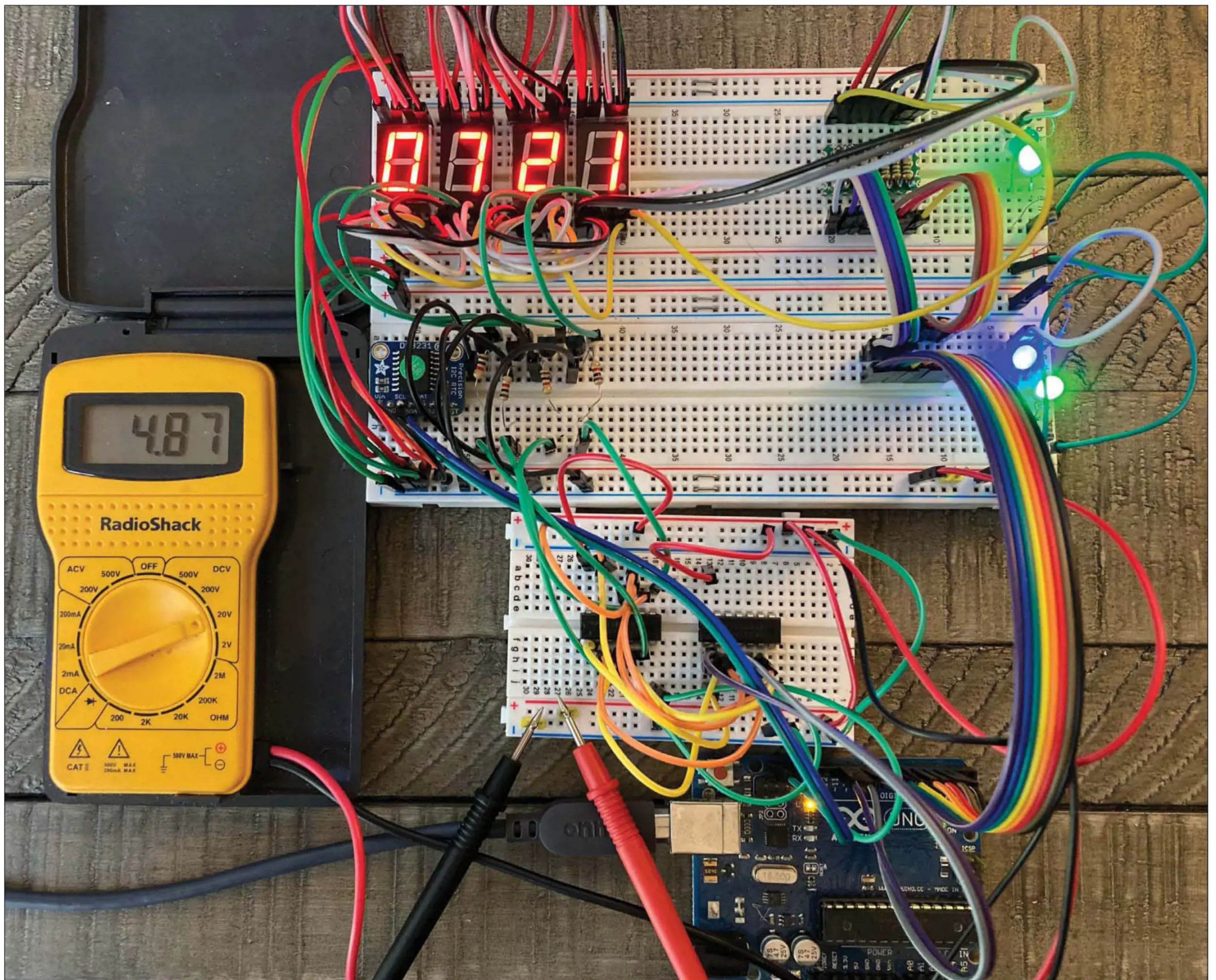


Fig.4. Working system with the 2:4 decoder on the half-size breadboard at the bottom.

As I said deep in the mists of time when we started *Part 1* of this *Arduino Bootcamp* series (*PE*, January 2023): ‘To prevent any such unfortunate occurrence, may I make so bold as to suggest you invest in an anti-static mat (<https://amzn.to/3g1YH4A>) and an anti-static wrist strap (<https://amzn.to/3WYZ9Bu>). Both typically come with crocodile clips (a.k.a. alligator clips in the US), which you can pull off to reveal banana plugs. My preferred *modus operandi* is to plug these banana plugs into an anti-static grounding plug (<https://amzn.to/3hDIcML>), which is – in turn – plugged into a wall socket or a power strip.’

With respect to the anti-static mat, get the largest one you can that fits your workspace (and your budget). Also, and this bit is *very important*, if you are working with static-sensitive components (diodes, transistors, ICs...), then *do not* waste your money on a blue or grey ‘Heat-Resistant Silicone Work Mat’ (like this one: <https://bit.ly/4bMwYfw>). I know the description includes the words ‘anti-static,’ but this

is naught but ‘click bait.’ Silicone is an electrical insulator, but anti-static mats must be able to conduct static build-up away to ground. A giveaway clue is that there is no way to connect these mats to ground!

Testing times

If you don’t already own a digital multimeter, now would be a really good time to invest in one. You have two main options in the form of hand-ranging or auto-ranging products. There are more pros and cons to this decision than you might suppose, but the accuracy will be the same in both cases.

For the purposes of this column, we will simply note that a hand-ranging multimeter takes a little longer to set up because you first need to select the desired range. This can be a little tricky for beginners to wrap their brains around. Suppose our hand-ranging multimeter indicates support for the 2V, 20V and 200V DC voltage ranges, for example (I can no longer remember why multimeter manufacturers use these arcane values). The 2V setting will measure voltages

up to 2V, the 20V setting will measure voltages up to 20V, and so on. Now suppose we wish to measure a signal that we know can vary only between 0V and 5V, like the signals generated by our Arduino Uno, for example. In this case, we would select the 20V range because this is the *closest* to the values we want to measure while still being *higher* than the values we want to measure. If we were to select the 200V range, this would still measure our signal, but with less precision.

By comparison, in the case of an auto-ranging multimeter, the setup is faster because all we need to do is select the DC voltage option. However, this type of multimeter will take longer to make the reading because it first needs to decide on the most appropriate range.

For what we are doing here, which will include verifying our power connections and monitoring the states on our logic signals, I would suggest buying a cheap-and-cheerful hand-ranging device, such as the ULTRICS Digital Multimeter, which is available for only £9.99 from Amazon at the time of writing: <https://bit.ly/4dMDu7V>

One very important point when you opt for a cost-conscious offering like this is to use it only for low-voltage and low-current work, say up to 40V and a couple of amps. *Don't* use it for mains or higher-voltage applications. I'm in no way impugning the purveyors of these products, but why take any chances?

If you decide to delve deeper into electronics, then at some stage in the future you may decide to invest in more sophisticated equipment, for example, an oscilloscope. I was just about to say that, since you are reading this *Arduino Bootcamp* column, you probably don't have a tool like this in your collection. But then it struck me that you could be an analogue hero dipping your toes in the digital waters for the first time, in which case (a) you may indeed have an oscilloscope and (b) welcome to the digital world; come on in, the water's fine.

That's logical

One very useful tool for the sort of work we're currently doing is a logic probe, which can be used to detect and reflect our logic 0 (0V) and logic 1 (5V) states. These are cheap, cheerful and extremely useful. You can buy one already built, like the Laser 5263 for £16.44 (<https://bit.ly/4aqwo6h>) or the JDXFENG for £3.99 (<https://bit.ly/3wSHovd>), or you might even opt for a DIY soldering kit version for £8.75 (<https://bit.ly/3yv5D3e>).

I decided a logic probe would be helpful in tracking down my own problem, but I can't remember where I put mine, and I didn't want to wait for one to be delivered (or spend any money on one, for that matter), so I decided to whip up a little something with what I had to hand. In a crunchy nutshell, we are talking about using a small breadboard (or part of our existing breadboard), two LEDs, two resistors, and a multi-core jumper wire. We will use items from the various kits we purchased in *Part 1* of this epic saga (*PE*, January 2023). A schematic is shown in Fig.5.

The reason I selected red and yellow LEDs is that the ones in my kit both claim forward voltage drops of 2V. I would

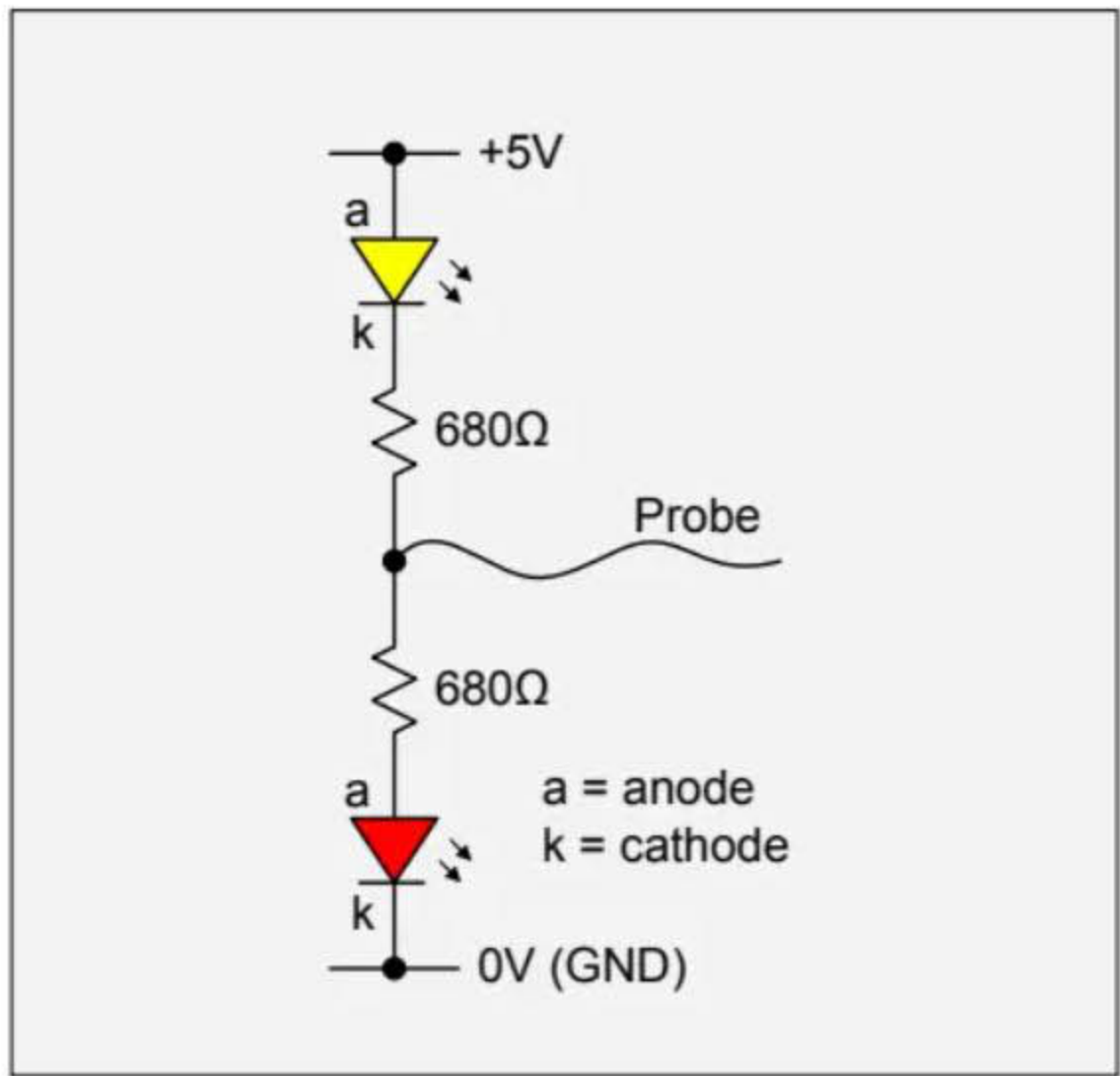


Fig.5. Logic tester.

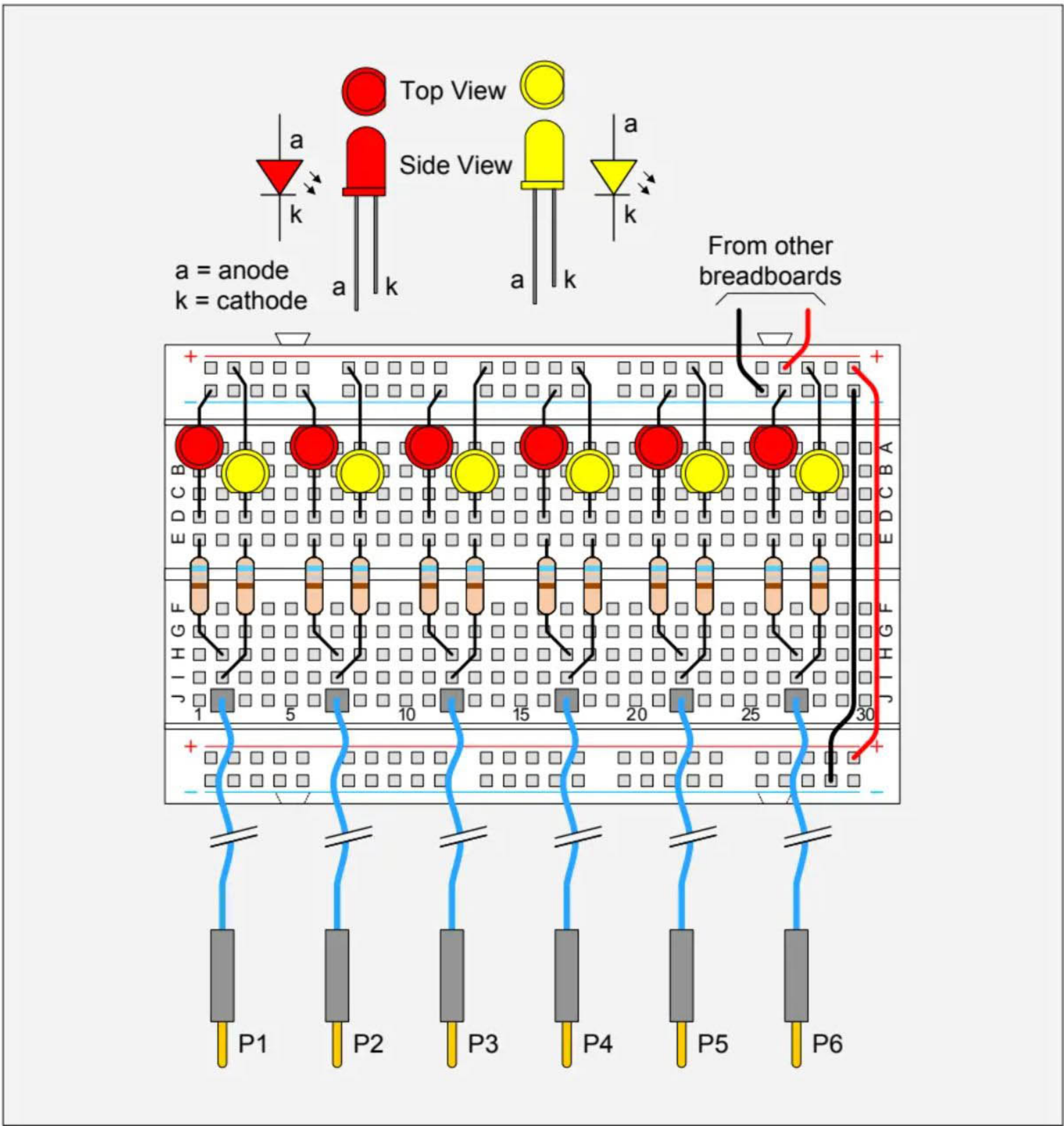


Fig.6. Six logic probes.

Components from Part 1

- LEDs (assorted colours) <https://amzn.to/3E7VAQE>
- Resistors (assorted values) <https://amzn.to/3O4RvBt>
- Solderless breadboard <https://amzn.to/3O2L3e8>
- Multicore jumper wires (male-male) <https://amzn.to/3O4hnxx>

Components from Part 2

- 7-segment display(s) <https://amzn.to/3Afm8yu>

Components from Part 5

- Momentary pushbutton switches <https://amzn.to/3Tk7Q87>

Components from Part 6

- Passive piezoelectric buzzer <https://amzn.to/3KmxjcX>

Components for Part 9

- SW-18010P vibration switch <https://bit.ly/46SfDA4>

Components for Part 10

- Breadboard mounting trimpots <https://bit.ly/3QAuz04>

Components for Part 12

- Light-Dependent Resistor <https://bit.ly/3S2430m>

Components for Part 13

- BC337 NPN Transistor <https://bit.ly/40xAgyS>

Components for Part 14

- HC-SR04 Ultrasonic Sensor <https://bit.ly/49AMBq4>

Components for Part 15

- Real-Time Clock (RTC) <https://bit.ly/3S90jHl>

Components for Part 18

- Long tailed (0.1-inch pitch) header pins <https://bit.ly/3U1Vp2z>

Components for Part 19

- Prototyping boards <https://bit.ly/3UMkcZ1>
- Kit of popular SN74LS00 chips <https://bit.ly/3wqgzyv>

Components for Part 20

- 16V 100µF electrolytic capacitors <https://bit.ly/44LzpNa>
- Ceramic capacitors (assorted values) <https://bit.ly/4bEAUiv>

have preferred to use red and green devices, but my green LEDs are noted as having a forward voltage drop of 3V.

Let's assume we leave the end of our probe waving in the air. In this case, once we've accounted for the two 2V voltage drops, we are left with $5V - 2V - 2V = 1V$ to power both LEDs. I've decided to use 680Ω resistors (with blue-grey-brown bands). Since there are two of them in series (one after the other), the total resistance is 1,360Ω. Using Ohm's law of $V = I \times R$, from which we derive $I = V / R$, we know our current (I) will be $1V / 1,360\Omega = \sim 0.7mA$. What this means is that both of our LEDs will glow very dimly indeed.

Now suppose we connect the flying end of our logic probe to +5V. This could be our power supply rail or an output from one of our digital ICs that's currently driving a logic 1 value. In this case, we will have 5V on both sides of our yellow LED, which therefore won't do anything at all. By comparison, our red LED now has the full 5V across it (and one 680Ω resistor). Removing this LED's 2V voltage drop gives $5V - 2V = 3V$. Since all the current is now flowing through a single resistor, this current will be $3V / 680\Omega = \sim 4.4mA$, which means the red LED will glow with a respectable brightness without hurting our eyes.

Similarly, if we connect the flying end of our logic probe to +0V (this could be the ground rail or an output from one of our digital ICs that's currently driving a logic 0 value, our red LED will be disabled and our yellow LED (which we are pretending is green) will glow like a champion.

As I previously mentioned, we could implement our logic probe on an unused portion of one of our existing

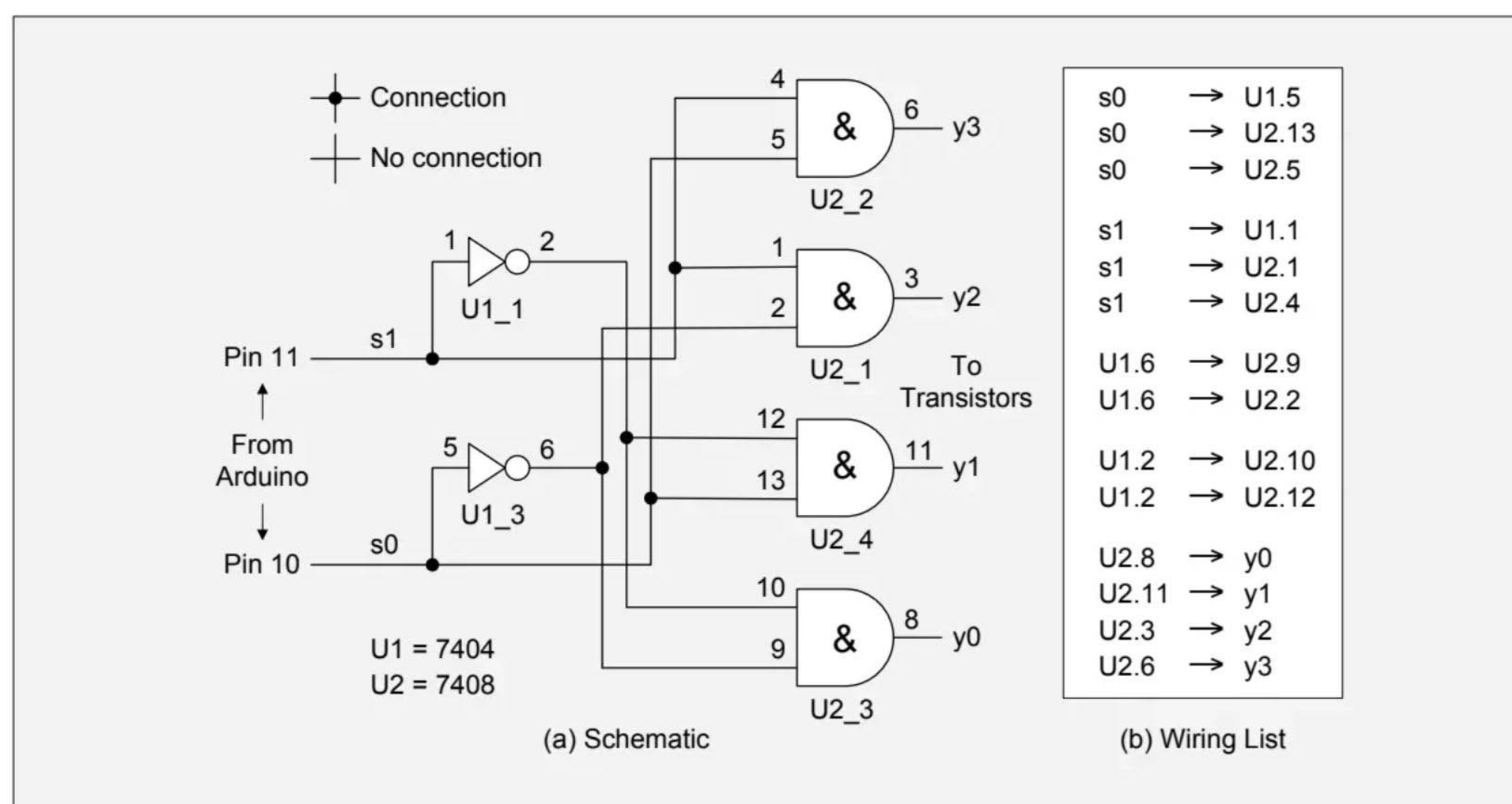


Fig.7. Schematic and wiring list for 2:4 decoder.

breadboards. However, if we have a spare board available – preferably a half-size board – then it might be better to employ this because it will be useful for other projects in the future. You can lay this out however you wish, but one possibility is shown in Fig.6.

Yes, I know, I got a bit carried away (LEDs... what can I say?). In the discussions below, you will be able to get by with a single logic probe, but more will be better, and they'll always come in handy in the future.

Note that, although I show the power and ground wires feeding our probe board as coming from our other breadboards, we could use a separate +5V supply if we wished. The important point here is that all the breadboards and power supplies must share (be connected to) the *same ground* (I'll explain the rationale behind this next month).

One point to go along with our philosophy of taking things step-by-step is to start by building only probe P1. Once you've added the components as shown, power things up and observe that both LEDs associated with P1 glow faintly. Next, plug the probe into the lower power rail and check that the red LED glows brightly while the yellow LED

is completely off. Finally, plug the probe into the lower ground rail and check that the yellow LED glows brightly while the red LED is completely off.

If any of this doesn't work as stated, start by using your multimeter to verify that you have 5V between the anode (a) of the yellow LED and the cathode (k) of the red LED. Next, check that you have both LEDs plugged in the right way round (compare the schematic to your breadboard layout). If all else fails, try swapping out the LEDs (they could be bad... remember ESD).

Once you have your first probe working (bravo!), build as many as you wish, one at a time, testing each one as you add it to your breadboard, and then proceed to the next section.

What we should have done

So, finally we come to what we should have done the first-time round. If you are having problems with your 2:4 decoder, then I know this is going to hurt, but let's power everything down, remove the 7404 and 7408 chips from

the breadboard (along with all their wiring), and we'll do it all again from the beginning.

Even if your decoder worked from the get-go (pat yourself on the back), you may still find these discussions to be instructive.

We start by returning to our 2:4 decoder schematic (Fig.7a) and using it to create a wiring list (Fig.7b). If you have a photocopier or a scanner handy, it would be a good idea to make a couple of copies of Fig.7.

The first check is to make sure our wiring list matches our schematic. Take a colored pencil and highlight a section of wire (not the whole wire) in the schematic, like the bit that goes from s0 to U1.5 (IC1, pin 5), and then highlight the corresponding line in the wiring list. Do this for all the wire segments in the schematic. When you've finished, all the wires in the schematic and all the lines in the list should be highlighted. If anything remains unhighlighted, we have a problem that needs to be resolved before we proceed further.

Before we do anything else, we want to deactivate our 7-segment displays to (a) make sure they don't flicker annoyingly and (b) make sure we don't have multiple

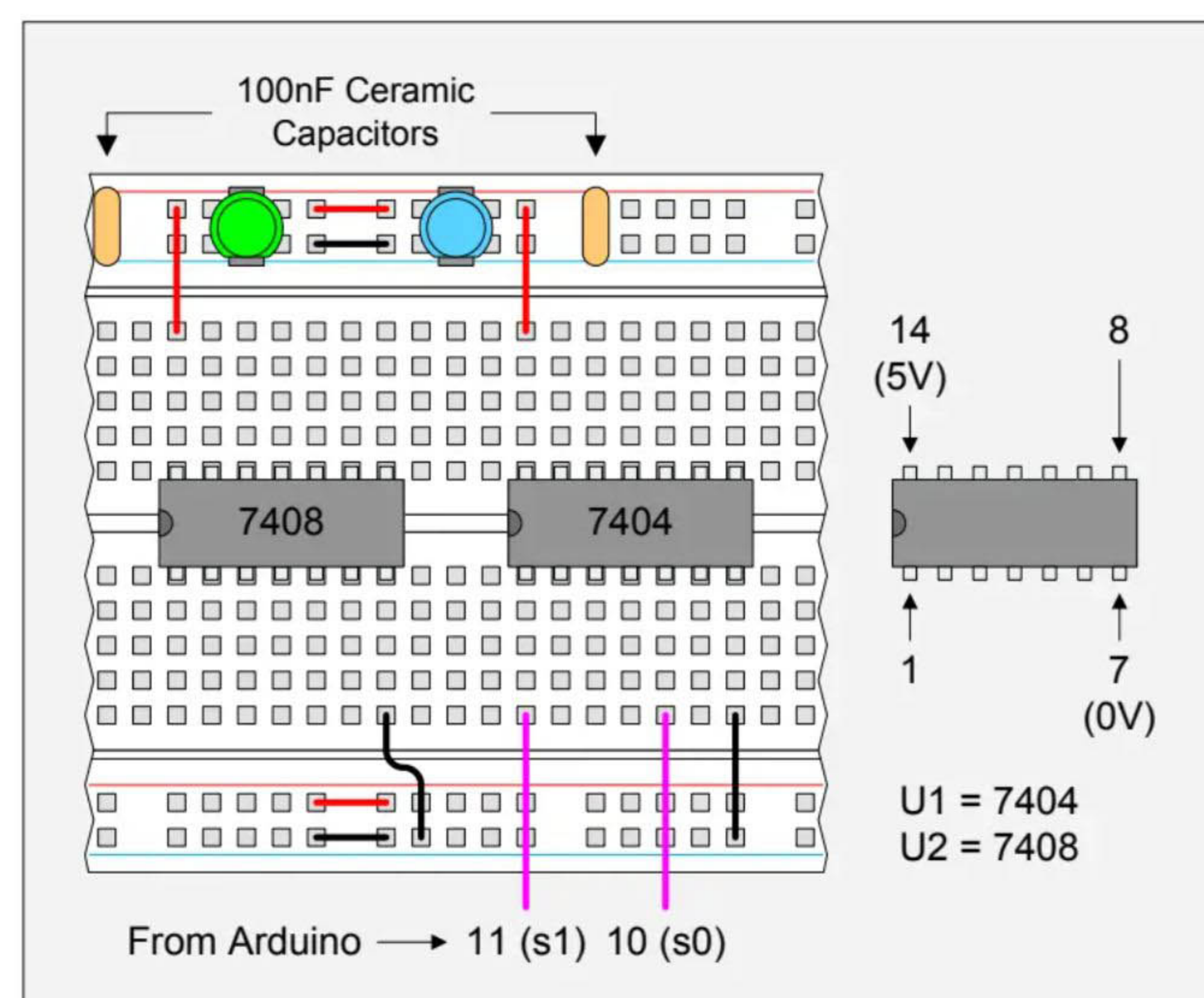


Fig.9. Add the ICs along with their power and ground connections and ceramic capacitors.

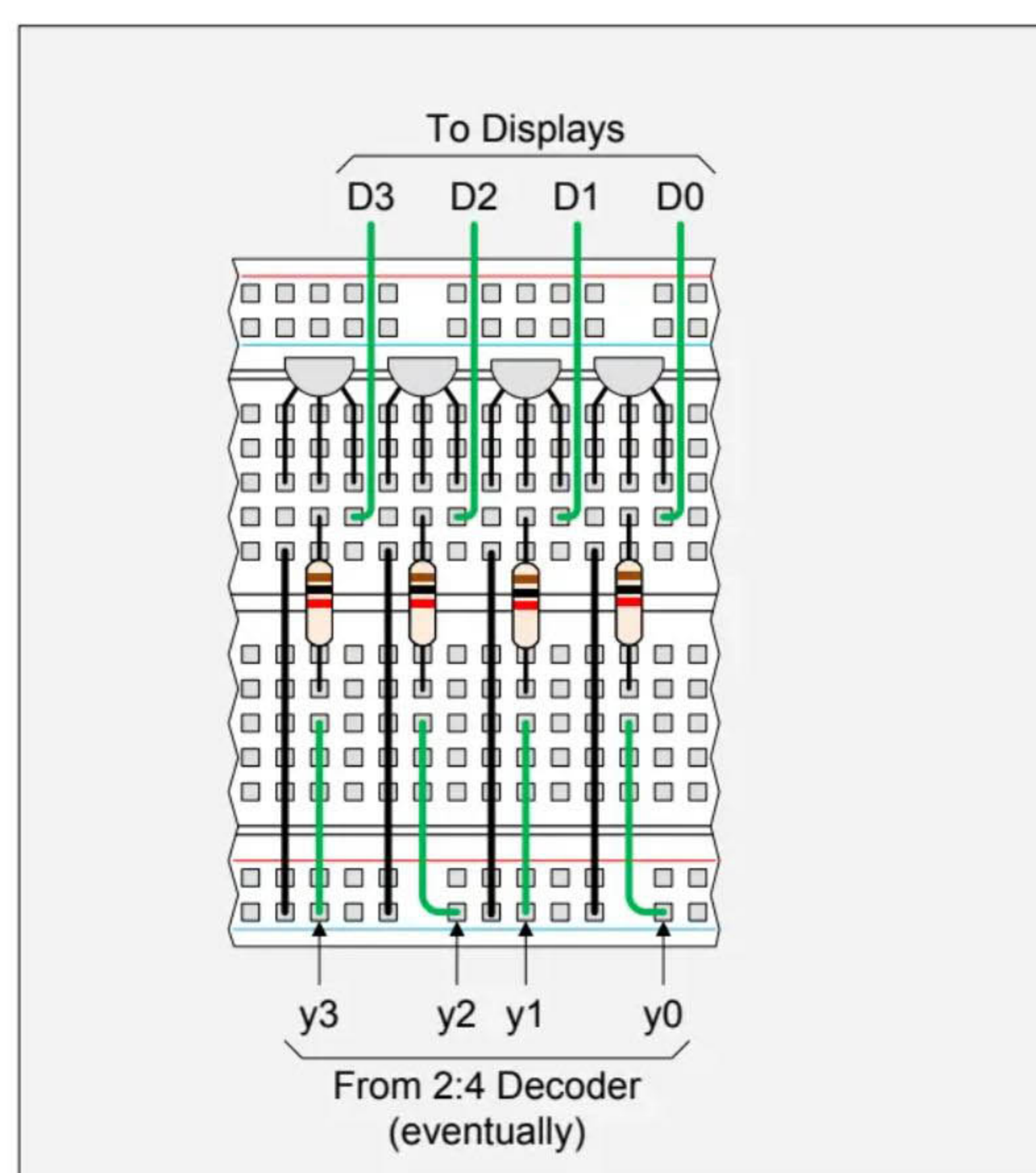


Fig.8. Disable the displays.

```

59 // Do this over and over again
60 void loop ()
61 {
62     // Activate each display in turn via 2:4 decoder
63     for (int iDisp = 0; iDisp < NUM_DISPLAYS; iDisp++)
64     {
65         DisplaySegs(ALL_SEGS_OFF);
66         digitalWrite(PinS0, iDisp & B0000001);
67         digitalWrite(PinS1, (iDisp >> 1) & B0000001);
68         DisplaySegs(DigitSegs[iDisp]);
69         delay(ON_OFF_TIME);
70     }
71 }

```

Listing 2a. Simple test of 2:4 decoder.

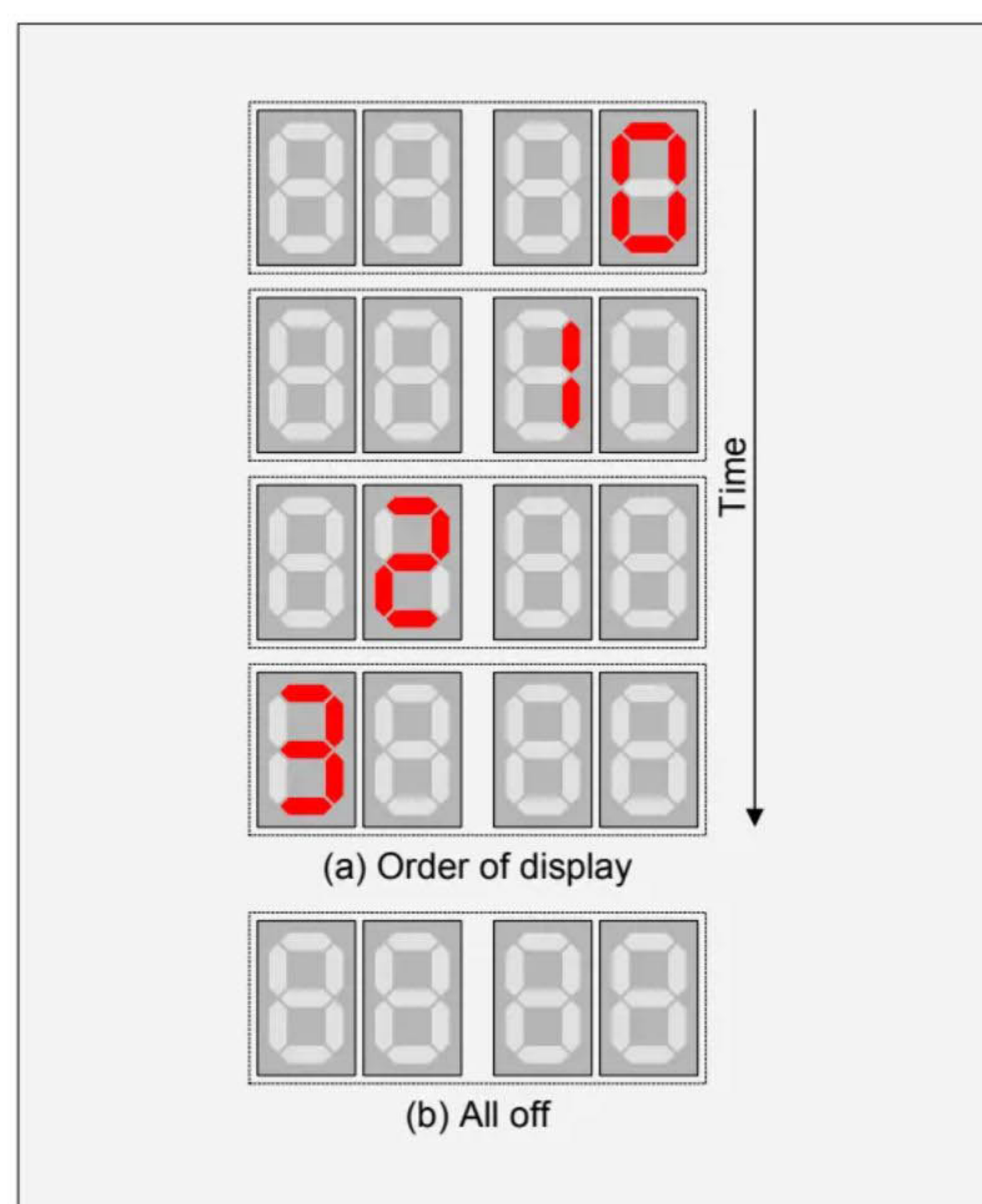


Fig.10. Test values.

displays active simultaneously because we know that's a bad idea. So, connect the ends of the y0, y1, y2 and y3 wires into the lower ground rail (the blue 0V rail) as illustrated in Fig.8. These are the wires connected to the resistors that drive the bases of our transistors. Connecting them to 0V will turn the transistors off, thereby disabling the displays.

Next, power everything up and verify that the green and blue LEDs straddling the power and ground rails at the top of the breadboard light up.

Now, set your multimeter to the 20V DC range, and then use its probes to check the voltage across the power and ground rails. As a 'rule-of-thumb,' I typically apply the black probe first. Start with the

pair of rails at the bottom of the board and then the pair at the top of the board.

Ideally, you should see a 5V (meaning +5V) value, but the Arduino's supply isn't ferociously robust, so you may have a slightly lower reading. In my case, I'm seeing 4.87V, as shown in Fig.4. If your reading is substantially lower than this – say 4.5V, for example – then you have a problem, possibly even a bad breadboard, which is what happened to me.

If you get a negative reading like -5V, then this probably means you've applied your probes the wrong way round. Your red probe should be on the power rail and your black probe should be on the ground rail. If the reading continues to show negative, then check the probe connections at the multimeter end. The black wire should be plugged into the 'COM' (common) port and the red wire should be plugged into the 'VΩmA' port.

Now, power everything down again, add the 7404 and 7408 ICs to the breadboard, and connect their power and ground pins as shown in Fig.9. Remember that the 'dimple' at one end of the package is used to determine which end is which. In this case, both of our chips use pin 14 for power and pin 7 for ground, but every IC is different, so you should always consult the data sheets for any chips you are using. While we're at it, let's also add the two 0.1μF ceramic capacitors we discussed earlier. We'll mount these as close to each IC's power pin as possible.

Feel the power!

Re-apply the power, and then check the voltage across pins 7 and 14 of the 7404. I mean this literally – apply the black probe to pin 7 and the red probe to pin 14. Once again, this should reflect a voltage of ~5V. Next, verify that the voltage across pins 7 and 14 of the 7408 is the same ~5V.

Testing, testing

This is where things get interesting. All we are going to do is repeatedly write the values 0, 1, 2 and 3 to the displays D0, D1, D2 and D3, respectively. Remember that we can have only one display active at a time. Also, since we want to be able to see what's happening, we'll set things up so each value will be on for one second and off for three seconds.

Let's start by creating a simple test program (file **CB-Aug24-02.txt**). I've based this on the latest and greatest program from the previous issue, but I've stripped out anything extraneous to what we're trying to do here. As an example, our `loop()` function is now as shown in Listing 2a.

With respect to the `for()` loop established on Line 63, the `iDisp` control variable repeatedly counts from 0 to 3.

We start by turning all our display segments off (Line 63). Then we set the values on our `s0` and `s1` signals (Lines 66 and 67). We derive these signals from the current value of `iDisp` (we discussed this last month). Next, we activate the segments associated with the current digit we wish to display (Line 68). Once again, we use the current value of

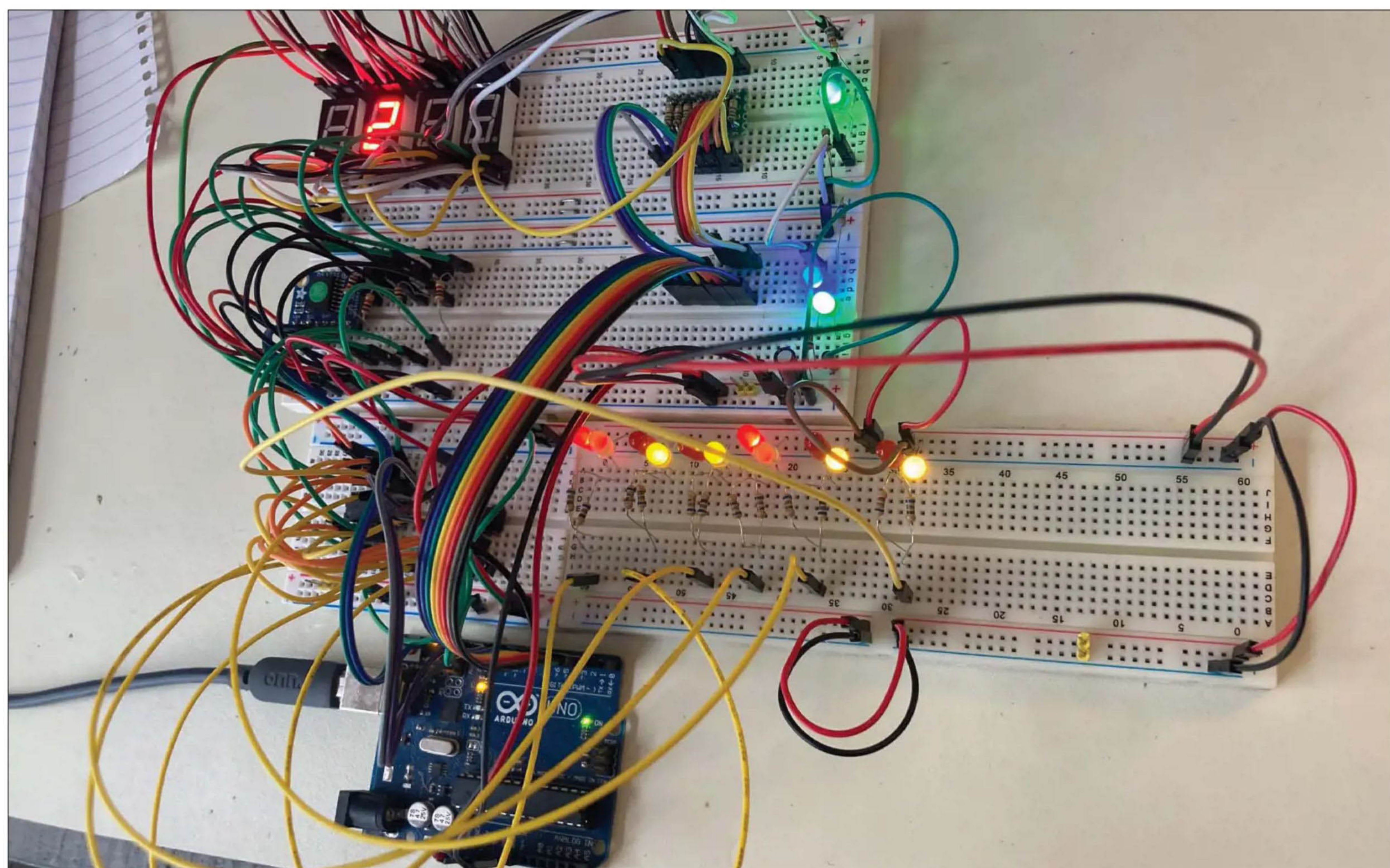


Fig.11. Using logic probes to make sure everything works.


```

31 byte DigitSegs[NUM_DIGITS+1] =
32 {
33     // ABCDEFGDP
34     B11111100, // 0
35     B01100000, // 1
36     B11011010, // 2
37     B11110010, // 3
38     B01100110, // 4
39     B10110110, // 5
40     B10111110, // 6
41     B11100000, // 7
42     B11111110, // 8
43     B11110110, // 9
44     ALL_SEGS_OFF
45 };

```

Listing 4a. Modifying the DigitSegs [] array.

iDisp to select the segments. Finally, we pause for a delay of ON_OFF_TIME, which I've set to 1000 milliseconds (1 second).

All of this means that when we run our program, the sequence presented on our displays will appear as shown in Fig.10a.

Well, the sequence would appear as shown in Fig.10a except for two small points (I'm reminded of Black Adder telling Baldrick, 'It's a cunning plan with one tiny flaw'). Can you spot the tiny flaws in our case? You're correct: (a) we haven't wired up our 2:4 decoder yet and (b) we've forcibly deactivated all our displays as per Fig.8. So, when we first run our program, the output will be as illustrated in Fig.10b.

Check that the two purple s1 and s0 wires are connected as shown in Fig.9. That is, one end of the s1 wire should be connected to the Arduino's digital pin 11, while the other end is connected to U1.1. Similarly, one end of the s0 wire should be connected to the Arduino's digital pin 19, while the other end is connected to U1.5.

We know that our s1 and s0 signals should be repeatedly going through the sequence 00, 01, 10 and 11. If we think about this another way, s0 will be going through the sequence 0, 1, 0, 1 (off for a second, on for a second, ...) while s1 will be going through the sequence 0, 0, 1, 1 (off for two seconds, on for two seconds, ...).

Assuming your multimeter is still set to 20V in the DC range, place the black probe on U1.7 (that's pin 7 of the 7404) and the red probe on U1.5. Since pin 5 is connected to signal s0 from the Arduino, you should see your multimeter display showing 0V, 5V, 0V, 5V. If not, re-check that the other end of the purple wire is plugged into pin 10 on the Arduino. If it is, try swapping the wire for a new one.

Once you've confirmed that you are seeing what you expect, highlight the s0 → U1.5 track segment in a new copy of Fig.7 (or use the old copy and mark it with a different colour). Do this both in the schematic (Fig.7a) and the wiring list (Fig.7b). Keep on doing this with every new segment as it's verified.

Leave the black probe on pin 7 and move the red probe to pin 1, which is connected to signal s1 from the Arduino. Now the multimeter display should show 0V, 0V, 5V, 5V. If not, re-check that the other end of the purple wire is plugged into pin 11 on the Arduino. If it is, try swapping the wire.

Online resources

For the purposes of this series, I'm going to assume that you are already familiar with fundamental concepts like voltage, current and resistance. If not, you might want to start by perusing and pondering a short series of articles I penned on these very topics – see: <https://bit.ly/3EguiJh>

Similarly, I'll assume you are no stranger to solderless breadboards. Having said this, even if you've used these little scamps before, there are some aspects to them that can trap the unwary, so may I suggest you feast your orbs on a column I wrote just for you – see: <https://bit.ly/3NZ70uF>

Last, but not least, you will find a treasure trove of resources at the **Arduino.cc** website, including example programs and reference documentation.

To be honest, digital multimeters aren't great for monitoring logic signals, but we can use them 'at a pinch.' You might be surprised to learn that analogue multimeters (or analogue voltmeters) are much more useful here. Given a choice, however, we will be better off using the simple logic probes we created earlier.

```

69 // Do this over and over again
70 void loop ()
71 {
72     int hour, minute;
73     int dispData[NUM_DISPLAYS];
74
75     DateTime now = rtc.now();
76
77     hour = now.hour();
78     minute = now.minute();
79
80     // Convert to 12-hour format
81     if (hour == 0)
82         hour = 12;
83     else if (hour > 12)
84         hour -= 12;
85
86     // Load HH:MM data into array
87     dispData[3] = hour / 10;
88     dispData[2] = hour % 10;
89     dispData[1] = minute / 10;
90     dispData[0] = minute % 10;
91
92     // Remove leading 0
93     if (dispData[3] == 0)
94         dispData[3] = NUM_DIGITS;
95
96     // Display HH:MM data
97     for (int iDisp = 0; iDisp < NUM_DISPLAYS; iDisp++)
98     {
99         DisplaySegs(ALL_SEGS_OFF);
100        digitalWrite(PinS0, iDisp & B00000001);
101        digitalWrite(PinS1, (iDisp >> 1) & B00000001);
102        DisplaySegs(DigitSegs[dispData[iDisp]]);
103        delay(ON_OFF_TIME);
104    }
105 }

```

Listing 4b. Modifying the loop () function.

If you created only one probe, connect it to U1.5 and watch the LEDs flash yellow, red, yellow, red, then connect it to U1.1 and watch the LEDs flash yellow, yellow, red, red. If you created two probes, you can monitor both signals simultaneously.

Now, let's perform our first test on U2 (the 7408). Add a wire to connect U1.5 (s0) to U2.5 and use one of your logic probes to verify that you are still seeing the yellow, red, yellow, red pattern on U2.5 (if not, check both ends are plugged into the right holes in the breadboard and/or swap out the wire). Add another wire to connect U1.1 (s1) to U2.4 and use one of your logic probes to verify that you are still seeing the yellow, yellow, red, red pattern on U2.4.

U2.4 and U2.5 are inputs to one of the AND gates in U2. If everything looks good, connect a probe to the output of this AND gate, which is U2.6. In this case, you should be seeing yellow, yellow, yellow, red.

This is the good part. Disconnect the end of the green y3 wire from the ground rail (Fig.8) and connect it to U2.6. Display D3 should present blank, blank, blank, 3. That is, it should be off for three seconds and then display the number 3 for one second and then repeat.

OK, now you know the drill. Use a similar process to bring up the remaining displays, one at a time. I only hope you will be as happy as I was once everything starts to work as planned (Fig.11)

All that remains now is to reload the program that uses our 2:4 decoder to control the displays and show the time (file CB-Aug24-03.txt).

And finally...

I hope you haven't forgotten our pop quiz from the beginning of this column. What we want to

do is modify our program to replace a 0 with a blank on display D3. As always, there are numerous ways to do this. The approach I opted to use works as follows (file CB-Aug24-04.txt).

First, as illustrated in Listing 4a, I modified the instantiation of our DigitSegs [] array on Line 31 by increasing its size from NUM_DIGITS (ten elements that are indexed from 0 to 9) to NUM_DIGITS+1 (eleven elements that are indexed from 0 to 10). This new element, which corresponds to all our segments being off, appears on Line 44. We can access this using DigitSegs [10] in our code. We could have specified this directly as B00000000, but I decided to use our existing ALL_SEGS_OFF definition.

Next, we need to add a simple test just before we display the digits. This test is shown on Lines 93 and 94 of Listing 4b. If the digit to be displayed on display D3 is 0, we replace this with NUM_DIGITS, which is defined as being 10. This will cause the call to DisplaySegs () on Line 102 to display our blank.

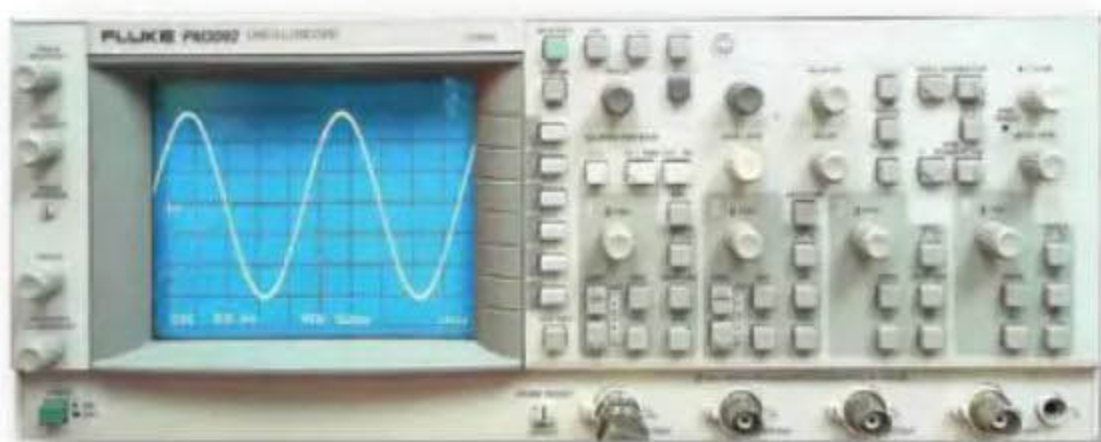
Next time

I know you're as tired of hearing me say this as I am of saying it, but in our next column we honest-to-goodness are going to start using a BCD-to7-segment decoder in our clock. As always, until that frabjous time comes (next month, I promise!), I welcome your comments, questions and suggestions.



Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technological geekdom.

Comments or questions? Email Max at: **max@CliveMaxfield.com**



Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250



HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

STEWART OF READING

17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: 0118 933 1111 Fax: 0118 933 2375
USED ELECTRONIC TEST EQUIPMENT
Check website **www.stewart-of-reading.co.uk**
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400	Marconi 2305	Modulation Meter	£250
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400	Marconi 2440	Counter 20GHz	£295
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900	Marconi 2945/A/B	Communications Test Set Various Options	POA
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA	Marconi 2955	Radio Communications Test Set	£595
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA	Marconi 2955A	Radio Communications Test Set	£725
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295	Marconi 2955B	Radio Communications Test Set	£800
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA	Marconi 6200	Microwave Test Set	£1,500
HP8903A/B	Audio Analyser	£750 – £950	Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
HP8757D	Scaler Network Analyser	POA	Marconi 6200B	Microwave Test Set	£2,300
HP3325A	Synthesised Function Generator	£195	Marconi 6960B	Power Meter with 6910 sensor	£295
HP3561A	Dynamic Signal Analyser	£650	Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
HP6032A	PSU 0-60V 0-50A 1000W	£750	Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350	Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
HP6624A	PSU 4 Outputs	£400	Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
HP6632B	PSU 0-20V 0-5A	£195	Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
HP6644A	PSU 0-60V 3.5A	£400	Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
HP6654A	PSU 0-60V 0-9A	£500	Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000	Farnell AP100-90	Power Supply 100V 90A	£900
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA	Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
HP83624A	Synthesised Sweeper 2 – 20GHz	POA	Racal 1991	Counter/Timer 160MHz 9 Digit	£150
HP8484A	Power Sensor 0.01-18GHz 3nW-10µW	£75	Racal 2101	Counter 20GHz LED	£295
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750	Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250	Racal 9300B	As 9300	£75
HP8566B	Spectrum Analyser 100Hz – 22GHz	£1,200	Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
HP8662A	RF Generator 10kHz – 1280MHz	£750	Solartron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325	Solartron SI 1255	HF Frequency Response Analyser	POA
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800	Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750	Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700	Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65



HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA



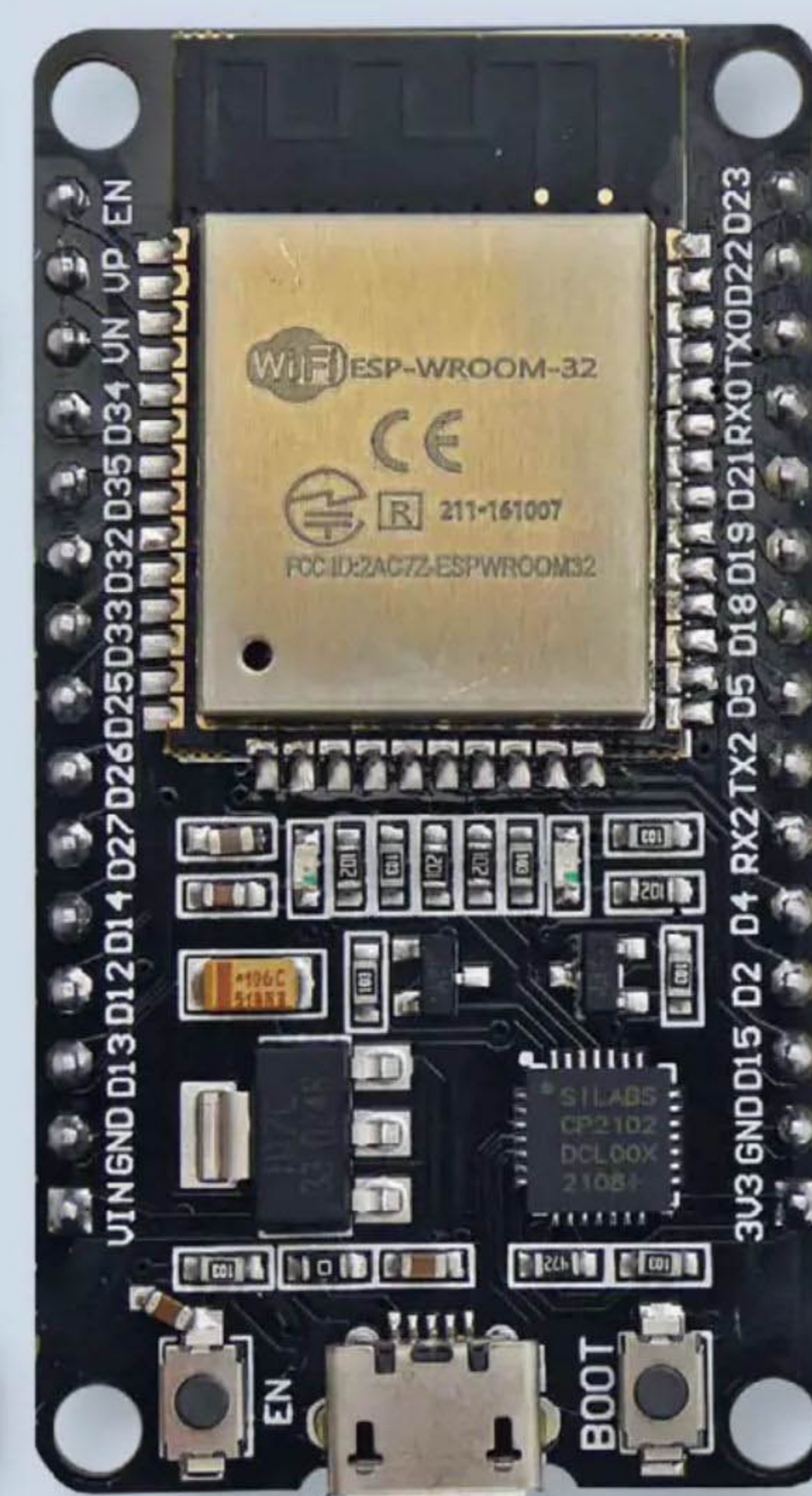
Marconi 2955B Radio
Communications Test Set – £800

Teach-In 2024

Learn electronics with the ESP32

by Mike Tooley

Part 6 – The ESP32 as a Wi-Fi Access Point



In the previous part of our Teach-In series, we introduced the 1-wire and I²C bus interface standards and showed how you can easily add a low-cost LCD display to your ESP32 projects. *Coding Workshop* introduced maths operations and functions and our *Teach-In Project* featured the design, construction and coding of a simple environmental monitor displaying temperature, barometric pressure and humidity.

This month, we begin to explore some of the ESP32's inbuilt features that help make wireless communication a reality. We begin by introducing some of the basic concepts and terminology associated with Wi-Fi technology and how the ESP32 can scan for available networks to which it can connect. Our *Practical Project* shows how the ESP32 can be configured for use as a Wi-Fi Access Point, making environmental data from a low-cost sensor available for remote display on a smartphone, tablet, laptop or desktop PC.

The learning objectives for the sixth part of our series are, know:

- How to set up and use ESP32 Wi-Fi
- The difference between Station (STA) mode and Access Point (AP) mode
- How to run a scan for nearby wireless networks
- How to configure the ESP32 Wi-Fi in AP mode
- How to apply simple Arduino C++ string handling techniques
- That HTML is used to display AP data in a web browser.

Getting to grips with Wi-Fi

Since this is the first time that we've used Wi-Fi in this *Teach-In* series it's worth starting by explaining a few of the concepts that underpin the technology. By enabling connectivity without the need for cabling, Wi-Fi is now commonly used in modern homes and businesses. All 'smart' devices make use of Wi-Fi and even a humble item such as a doorbell now employs this technology. We will start by explaining some key terms before moving on to explain the two modes of connection that the ESP32 uses.

Internet Protocol (IP)

First proposed in 1974, Internet Protocol (IP) is simply a set of rules that facilitates the exchange of data between devices connected to a network. IP defines the structure of the packets that encapsulate the data to be delivered from a source to a host, and vice versa. Each packet consists of the data to be transmitted together with a header that incorporates an IP address. IP forms part of the layered protocol often referred to as 'Internet protocol suite', the different levels of which are categorised as Application, Transport, Internet and Link.

About Teach-In

Our latest *Teach-In* series is about using the popular ESP32 module as a basis for learning electronics and coding. We will be making no assumptions about your coding ability or your previous experience of electronics. If you know one but not the other, you have come to the right place. On the other hand, if you happen to be a complete newbie there's no need to worry because the series will take a progressive hands-on approach. There will be plenty of time to build up your knowledge and plenty of opportunity to test things out along the way.

We've not included too much basic theory because this can be easily found elsewhere, including several of our previous *Teach-In* series, see:

<https://bit.ly/pe-ti>

<https://bit.ly/pe-ti-bundle>

Each month, there'll be projects and challenges to help you check and develop your understanding of the topics covered.

IP addresses

For reasons that will be obvious, every device connected to a network needs to have its own unique address. This address is simply a 32-bit (IPv4) or 128-bit (IPv6) number. These are usually expressed as a sequence of bytes with the decimal values quoted for each. Here's an example of a four-byte *local* IP address: 192.168.1.1. Note that devices connected to your own router are all members of the same network and can

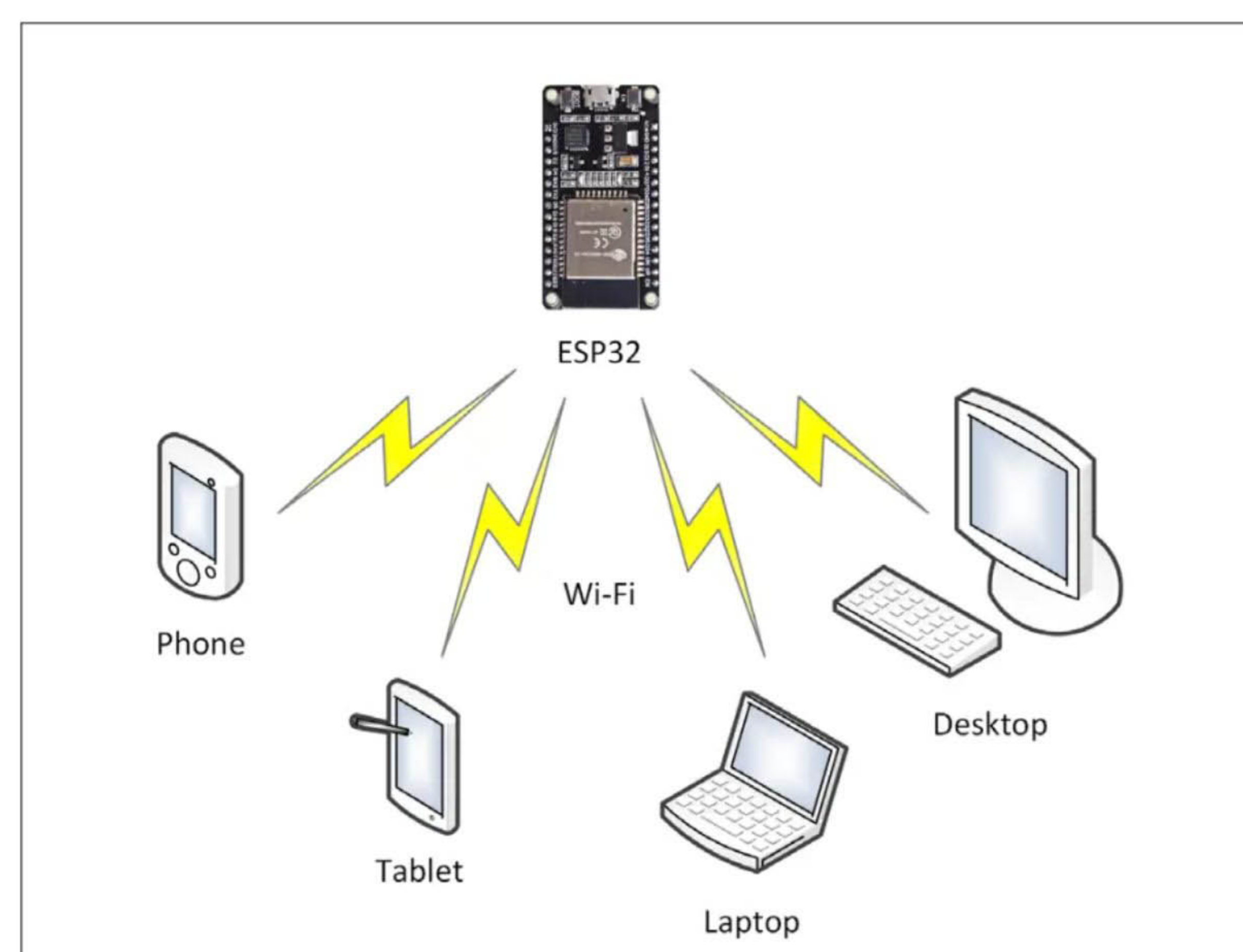


Fig.6.1. An ESP32 operating in Access Point (AP) mode.

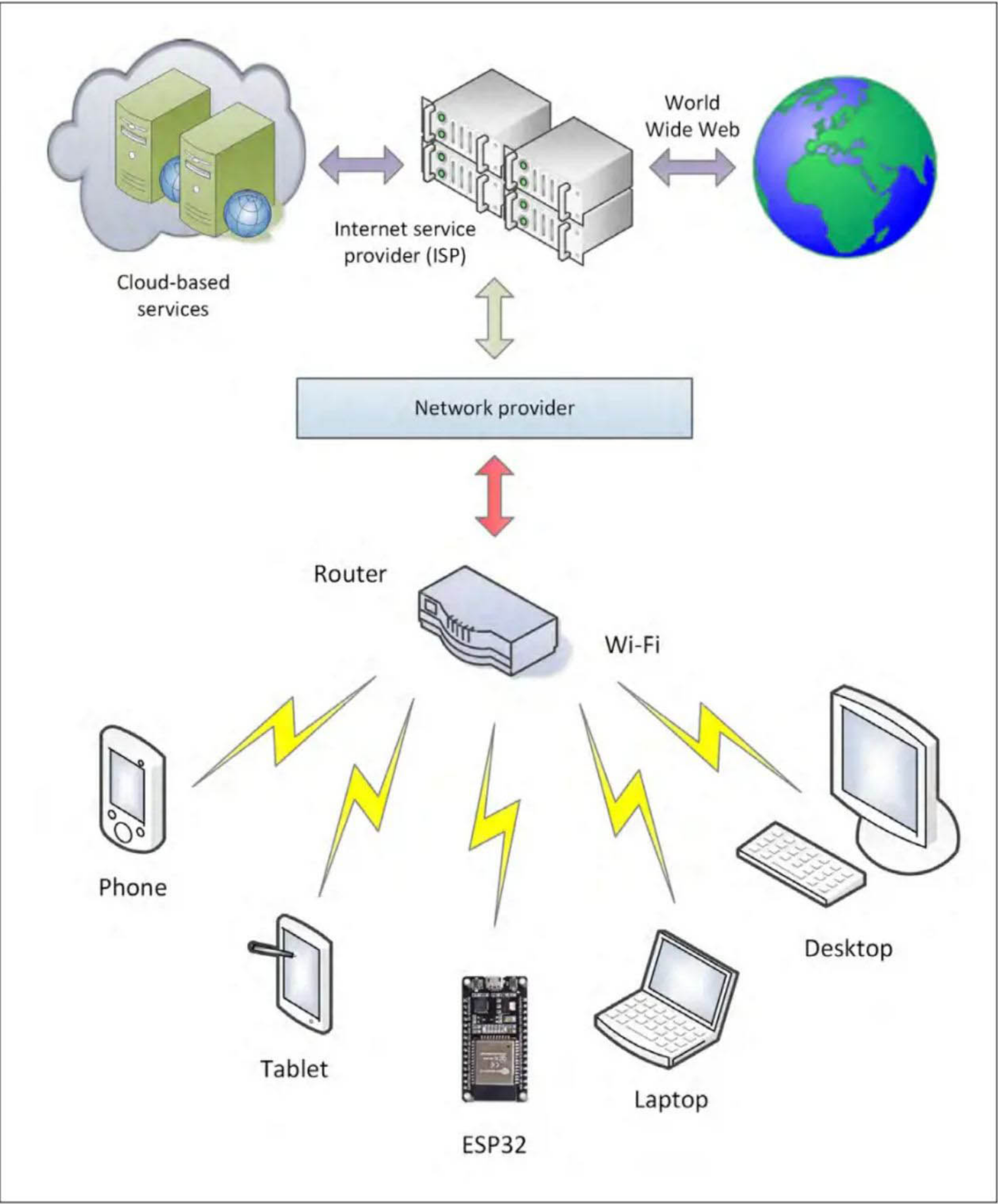


Fig.6.2. An ESP32 operating in Station (STA) mode.

communicate using these local IP addresses without the need for an Internet connection.

Dynamic Host Control Protocol (DHCP)

IP addresses within the network are assigned by the router using Dynamic Host Control Protocol (DHCP). This process is automatic, and it occurs whenever a device first connects to the network. The router then updates its list of connected devices together with their allocated IP addresses. If you have the correct network privileges you can easily log into the router and view this list.

Wi-Fi bands

Wi-Fi communication operates in two discrete frequency bands. The original 2.4GHz band and the more recently available 5GHz band. Each of these bands is further subdivided into several channels, each with an allocated number. Most existing

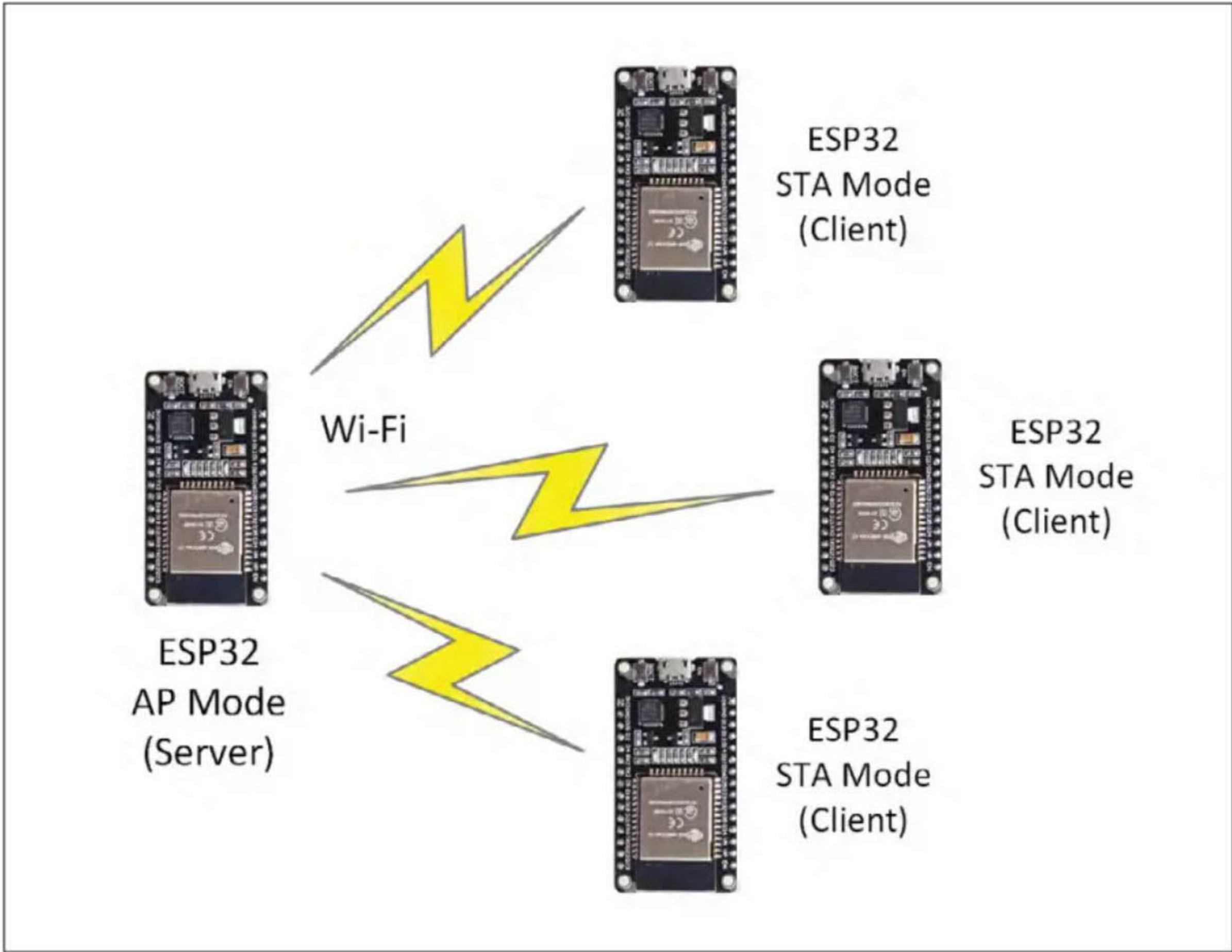


Fig.6.3. An ESP32 serving data to several ESP32 clients. versions of the ESP32 operate on the 2.4GHz band, but dual-band versions (such as the ESP32-C5) have recently become available. The 5GHz band suffers from less congestion and can therefore be more reliable, but unfortunately it's less robust and the range can be somewhat limited due to signal absorption and attenuation within buildings. For most purposes single-band operation at 2.4GHz is perfectly adequate for the ESP32.

Service Set Identifier (SSID)

An SSID is the name used to identify a Wi-Fi network and just comprises a sequence of alphanumeric characters. SSID is not case-sensitive and a default SSID is normally set by the equipment manufacturer and often appears printed on a label attached to the bottom or rear panel of a wireless router. The SSID can usually be changed easily if required.

```
Starting scan ...
Scan completed!
12 networks found
```

No.	SSID	RSSI	CH	Encryption
1	BTWholeHome-WRS3	-54	1	WPA2
2	vodafoneF159E9	-60	11	WPA2
3	BTWholeHome-WRS3	-67	1	WPA2
4	SKYPDTMK	-70	6	WPA2
5	BTWholeHome-WRS3	-74	1	WPA2
6	SKYPDTMK	-89	11	WPA2
7	BT-G2AGGC	-89	11	WPA2
8	EE WiFi	-89	11	open
9	TALKTALK0369A2	-92	1	WPA+WPA2
10	Methley-X2C2NC	-96	1	WPA2
11	SKYPDTMK	-96	11	WPA2
12	BT-PRFK5J	-97	11	WPA2

Fig.6.4. Typical output from Listing 6.1.

Table 6.1 Summary of common Wi-Fi encryption methods

Encryption	Outline
Wired Equivalent Privacy (WEP)	Outdated but still widely used. Available in 64, 128 and 256-bit versions. Officially retired by the Wi-Fi Alliance in 2004.
Wi-Fi Protected Access (WPA)	Formally adopted in 2003, the most-common WPA configuration is WPA-PSK (Pre-Shared Key). The keys used by WPA are 256-bit, a significant increase over the 64-bit and 128-bit keys used in the WEP system. WPA originally employed Temporal Key Integrity Protocol (TKIP) but this was later superseded by the Advanced Encryption Standard (AES).
Wi-Fi Protected Access II (WPA2)	Officially superseded WPA in 2006. It uses AES algorithms and CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) as a replacement for TKIP which is still available in WPA2 (as a fallback) and to ensure interoperability with WPA.
Wi-Fi Protected Access III (WPA3)	First announced in 2018 and offering significant improvements over WPA2. It uses Protected Management Frames (PMF) to enable encryption of the data used for network management (in addition to that used for the data itself). Also incorporates other security enhancements and protection against brute force and offline attacks. WPA3 192-bit Enterprise Mode currently offers the most advanced Wi-Fi encryption.

Listing 6.1 A simple ESP32 Wi-Fi scanner

```
/* Simple Wi-Fi Network Scanner */

#include "WiFi.h"

void setup() {
    Serial.begin(115200);

    // Initialise WiFi
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(1000); // Short delay
}

void loop() {
    Serial.println("Starting scan ...");
    int n = WiFi.scanNetworks(); // n is number of networks discovered
    Serial.println("Scan completed!");
    if (n == 0) {
        Serial.println("No networks discovered!"); // Alert if none found.
    } else {
        Serial.print(n);
        Serial.println(" networks found");
        Serial.println("No. \tSSID\t\t\t\t\tRSSI\tCH\tEncryption");
        for (int i = 0; i < n; ++i) {
            // Print SSID and RSSI for each network found
            Serial.printf("%2d", i + 1);
            Serial.print("\t");
            Serial.printf("%-32.32s", WiFi.SSID(i).c_str());
            Serial.print("\t");
            Serial.printf("%4d", WiFi.RSSI(i));
            Serial.print("\t");
            Serial.printf("%2d", WiFi.channel(i));
            Serial.print("\t");
            switch (WiFi.encryptionType(i)) {
                case WIFI_AUTH_OPEN:
                    Serial.print("open");
                    break;
                case WIFI_AUTH_WEP:
                    Serial.print("WEP");
                    break;
                case WIFI_AUTH_WPA_PSK:
                    Serial.print("WPA");
                    break;
                case WIFI_AUTH_WPA2_PSK:
                    Serial.print("WPA2");
                    break;
                case WIFI_AUTH_WPA_WPA2_PSK:
                    Serial.print("WPA+WPA2");
                    break;
                case WIFI_AUTH_WPA2_ENTERPRISE:
                    Serial.print("WPA2-EAP");
                    break;
                case WIFI_AUTH_WPA3_PSK:
                    Serial.print("WPA3");
                    break;
                case WIFI_AUTH_WPA2_WPA3_PSK:
                    Serial.print("WPA2+WPA3");
                    break;
                case WIFI_AUTH_WAPI_PSK:
                    Serial.print("WAPI");
                    break;
                default:
                    Serial.print("Not known");
            }
            Serial.println();
            delay(10);
        }
        Serial.println("");
        WiFi.scanDelete(); // Tidy up
        delay(5000); // Delay before the next scan
    }
}
```

Gotcha!

To avoid making your network vulnerable and prevent an 'Evil Twin' attack, it is considered good practice to change your router's SSID. You will find further information in the documentation supplied with your router.

Received Signal Strength Indicator

RSSI provides you with an indication of the strength of a Wi-Fi signal. The greater the RSSI the better the signal quality with less susceptibility to errors. Strictly speaking, RSSI is a relative measurement whereas 'dBm' is an absolute measure of the power received from a Wi-Fi antenna. Despite this, RSSI is normally expressed in 'dBm' (decibels relative to 1mW) and a value of -70dBm (or greater) is often quoted as an indicator of good Wi-Fi performance. Note, however, that values in the range -70dBm to -90dBm can still prove satisfactory provided there is no co-channel contention.

Security and encryption

Since the introduction of Wi-Fi in the late 1990s, Wi-Fi has always had some form of security algorithm to provide for user authentication and communication encryption. Several different forms of data encryption help make networks more secure. Some that you are most likely to encounter are listed in Table 6.1.

In order of effectiveness, Wi-Fi encryption methods are usually ranked in the following order:

1. WPA3 (currently the most effective security)
2. WPA2 plus AES
3. WPA plus AES
4. WPA plus AES with fallback to TKIP
5. WPA plus TKIP
6. WEP
7. Open network (no security at all).

ESP32 connection modes

The ESP32's Wi-Fi API (Application Programming Interface) provides support for the industry standard IEEE 802.11b/g/n protocol driver. So, when connected to a network the ESP32 can be configured either in Access Point (AP) mode (Fig.6.1) or in Station (STA) mode (Fig.6.2).

Gotcha!

The negative values that are used when quoting RSSI simply indicate that the received signal power is below 1mW. Therefore, large *negative* values of RSSI correspond to weaker signals. You should also note that the decibel (dB) measure is logarithmic so that, for example, a power difference of 20dB actually corresponds to a ratio of 100:1.

Gotcha!

It's easy to confuse different Wi-Fi configurations. In AP Mode the ESP32 provides its own Wi-Fi network to which other devices connect, but in STA Mode the ESP32 just connects to an AP provided by another device, usually a wireless router.

In STA mode the ESP32 connects to another Wi-Fi network such as that provided by a nearby wireless router. In this mode the ESP32 effectively becomes a Wi-Fi client. Note that, whereas the AP to which the ESP32 connects is usually a router, an AP could also be provided by another ESP32 acting as a server.

When operating in AP Mode, Wi-Fi-enabled devices (such as a smartphone, tablet or PC) can connect to the ESP32. In this situation, the ESP32 creates its own Wi-Fi network to which other Wi-Fi devices can connect. In this mode, the ESP32 serves HTTP data that can be read and displayed using a Web browser running on a client device (Fig.6.3)

Check it out!

The time has come to make use of the ESP32's Wi-Fi capability, so let's get started by carrying out a scan to discover the available networks. Depending on where you live this can be as few as one (the Wi-Fi network provided by your own wireless router) to well over a dozen! Each network will display a different SSID and each will have a particular encryption method.

We need to start by initialising the ESP32's Wi-Fi in Station Mode (STA) using a line of the form:

```
WiFi.mode(WIFI_STA); // Initialise  
in STA mode
```

However, if by chance, the ESP32 has already been connected to an Access Point (AP) we first need to disconnect from it by using:

```
WiFi.disconnect(); // Disconnect  
from an AP
```

Then we need to scan the available networks, and determine the SSID and encryption method for each one. We will send the results using the ESP32's serial print function so that they can be

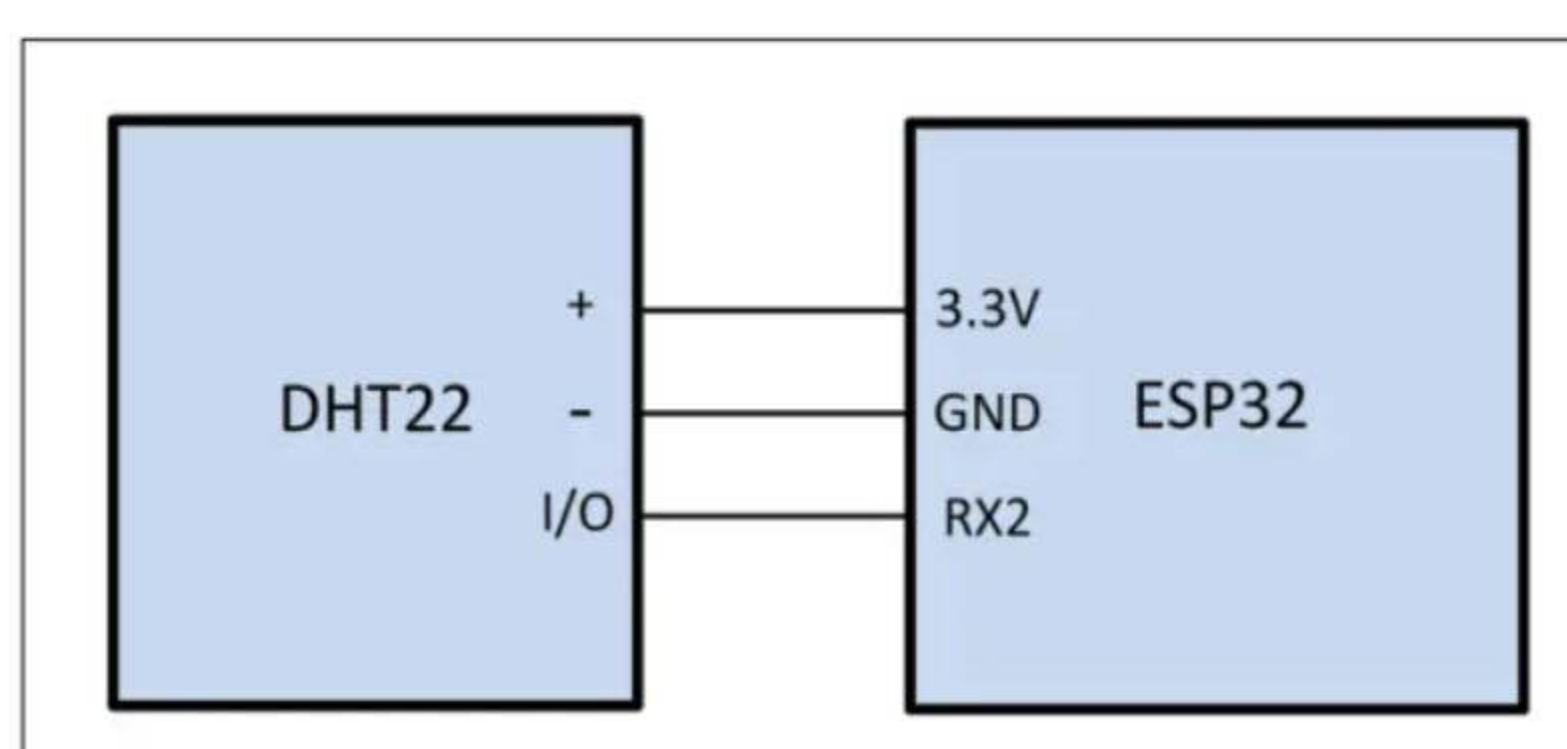


Fig.6.5. ESP32 and DHT22 interconnection.

Listing 6.2 Code for this month's Practical project

```
/*  
Basic WiFi Access Point with an ESP32 sending  
temperature and humidity data obtained from a  
DHT22 sensor. Uses the DHT22 library file by dvarrel  
*/  
  
#include <WiFi.h>  
#include <WebServer.h>  
#include <DHT22.h>  
  
// Enter your chosen SSID and password here  
const char* ssid = "ESP32";  
const char* password = "12345678";  
  
// Create an instance of the DHT22 class and  
// set the digital I/O to pin-21  
DHT22 dht22(21);  
  
// Configure the IP address  
IPAddress local_ip(192, 168, 1, 1);  
IPAddress gateway(192, 168, 1, 1);  
IPAddress subnet(255, 255, 255, 0);  
WebServer server(80);  
  
void setup() {  
    // Use the serial monitor  
    Serial.begin(115200);  
    // Configure WiFi connection  
    WiFi.softAP(ssid, password);  
    WiFi.softAPConfig(local_ip, gateway, subnet);  
    delay(100);  
    // Enable the AP server  
    server.on("/", handle_OnConnect);  
    server.onNotFound(handle_NotFound);  
    server.begin();  
    Serial.println("HTTP AP started");  
}  
  
void loop() {  
    server.handleClient();  
}  
  
void handle_OnConnect() {  
    Serial.println("Reading data...");  
    // Read the DHT22 sensor  
    float humidity = dht22.getHumidity();  
    float temperature = dht22.getTemperature();  
    server.send(200, "text/html", SendHTML(temperature, humidity));  
}  
  
void handle_NotFound() {  
    server.send(404, "text/plain", "Not found");  
}  
  
String SendHTML(float temperature, float humidity) {  
    String ptr = "<!DOCTYPE html> <html>\n";  
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";  
    ptr += "<title>ESP32 Access Point</title>\n";  
    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";  
    ptr += "body{margin-top: 50px;} h1 {color: #006600;margin: 50px auto 30px;} h2 {color: #660099;margin-bottom: 50px;}\n";  
    ptr += "</style>\n";  
    ptr += "</head>\n";  
    ptr += "<body>\n";  
    ptr += "<h1>ESP32 Access Point</h1>\n";  
    ptr += "<h2>Temperature: " + String(temperature) + "&deg;C</h2>\n";  
    ptr += "<h2>Humidity: " + String(humidity) + "%</h2>\n";  
    ptr += "</body>\n";  
    ptr += "</html>\n";  
    return ptr;  
}
```


Gotcha!

When exploring your network environment, it's always worth scanning several times to see if anything has changed. Wi-Fi networks are dynamic, and you will find that some will appear and disappear sporadically from the network list.

displayed using the IDEs and display them on our host PC. Since a further scan might reveal something different we will repeat this process every five seconds. Each time deleting the results of the previous scan in order to free memory. This is achieved using:

```
WiFi.scanDelete(); // Tidy up
and free some memory
```

The code for our simple Wi-Fi scanner is shown in Listing 6.1.

Fig.6.4 shows typical output produced by Listing 6.1. It's worth taking a close look at this. The Wi-Fi networks found are displayed in strict order of received RSSI (strongest at the top of the list). If you look at the RSSI column you will see that the best signal (at -54dBm) is received from one of the stations on the BTWholeHome network while the weakest (at -97dBm) is being received from the BT-PRFK5J network. You should also note that five of the networks are operating on Channel 11, three on Channel 1, and one on Channel 6. Note that BTWholeHome is a mesh network with three separate connecting points, one on each floor of the house. Fig.6.4 also reveals that the most common encryption method is WPA2, with one network using WPA+WPA2, and another is 'open' (with no encryption).

Practical project

We mentioned earlier that in AP mode the ESP32 can create its own Wi-Fi network to which other smart devices can be connected. This allows you to monitor and view data remotely without the need for a wireless router or host computer. To show how useful this is we've created an application based on the DHT22 temperature and humidity sensor that we met last month.

The complete code for our *Practical Project* is shown in Listing 6.2. Note that

Gotcha!

You might have noticed that we've used '\t' in several of the `Serial.println` statements of Listing 6.1. The \t simply inserts a non-printing tab character in the string to be printed and this can be used to ensure that the text is aligned neatly.

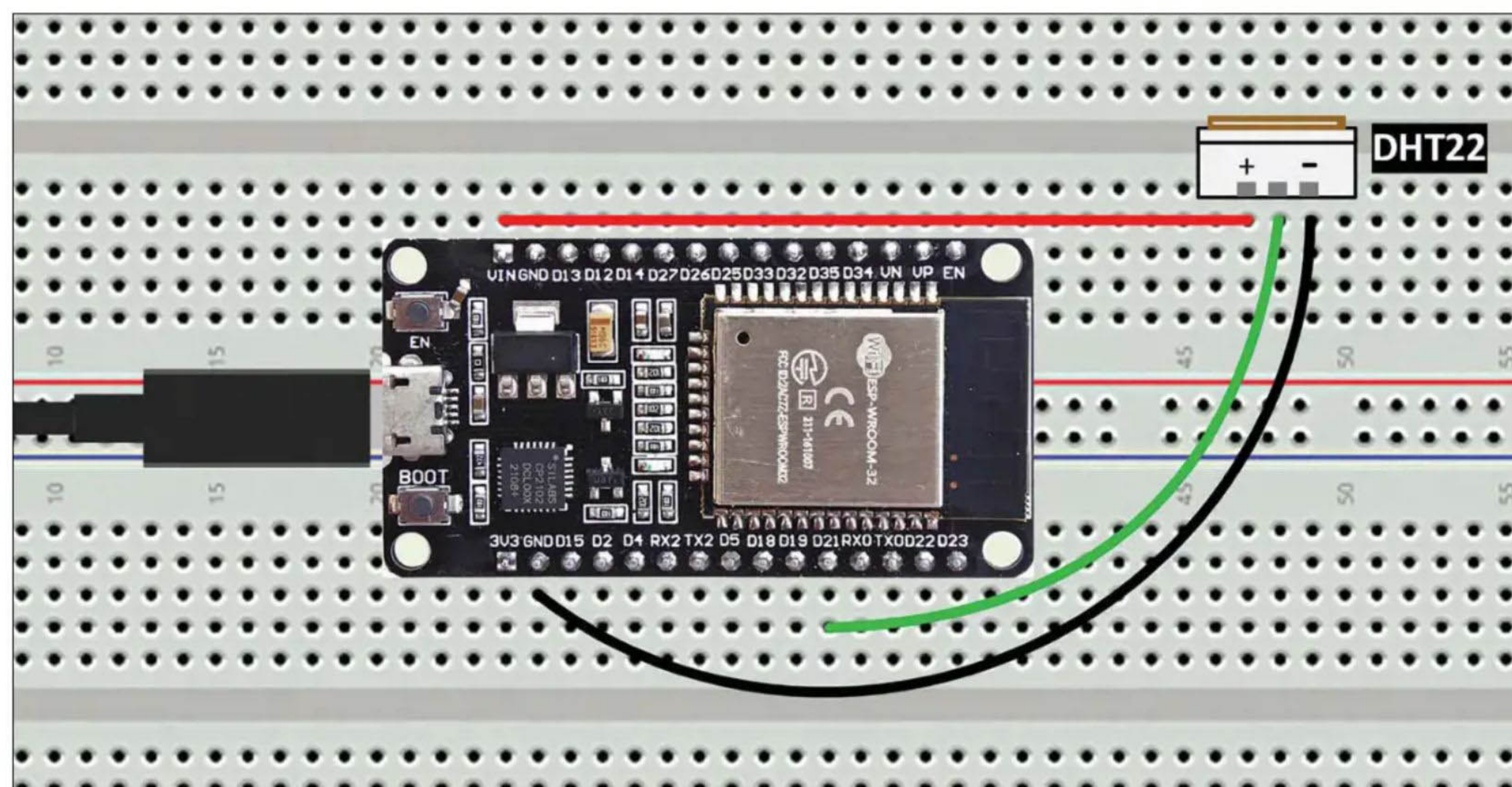


Fig.6.6. Layout diagram for Fig.6.5.

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0
HTTP AP started
```

Fig.6.7. Serial monitor output from Listing 6.2.

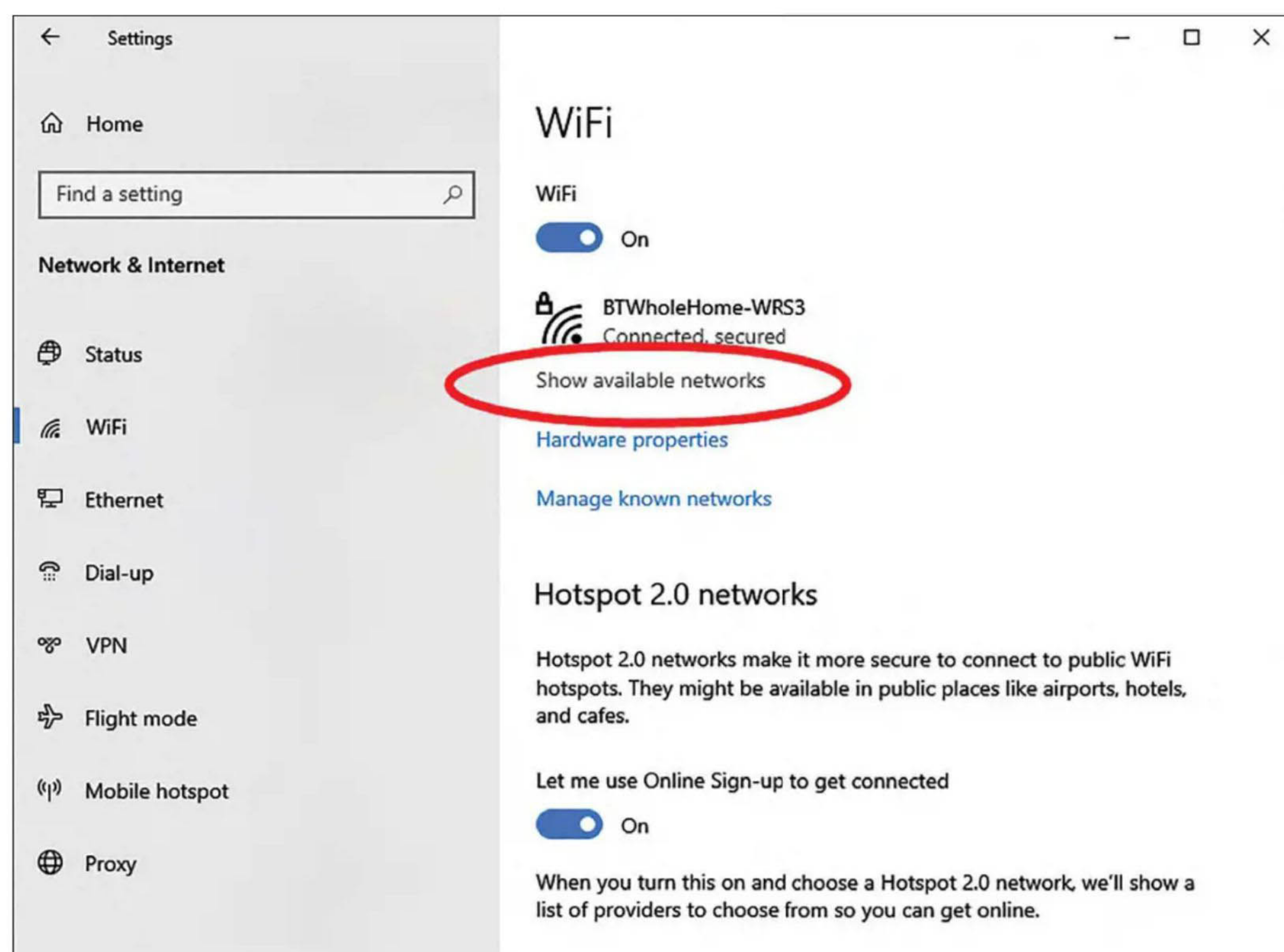


Fig.6.8. Typical Wi-Fi configuration screen on a desktop PC.

we need to include three libraries at the beginning of the code. This is done with the following lines:

```
#include <WiFi.h> // To use
ESP32 Wi-Fi
```

```
#include <WebServer.h> // To
use the ESP32 as a web server
```

```
#include <DHT22.h> // To use
the DHT22 sensor
```

It may not be necessary to download and install all of these library files as you

might already have them. If not, just go to the IDE's Library Manager (as described in earlier parts of this *Teach-In* series).

In Listing 6.2 we have included several lines of code that are needed to configure the IP address used by the ESP32. Note that we've set the local IP address to 192.168.1.1. This is the address that you will need to enter into your browser's address bar to access the hypertext transfer language (HTML) data served by the ESP32.

Rather than just display plain text, we've included some HTML tags to improve the appearance of the data.

Listing 6.3 Some examples of string handling

```
/* Some simple examples of string handling */
#include <string>
void setup() {
    Serial.begin(115200);
}

void loop() {
    Serial.println("String handling examples ...");
    char firstString[] = "Practical Electronics";
    int n = strlen(firstString);
    Serial.print("First string length = ");
    Serial.println(n);
    // Print the first string
    Serial.println(firstString);
    // Print the first character
    Serial.print("First character = ");
    Serial.println(firstString[0]);
    // Print the last character
    Serial.print("Last character = ");
    Serial.println(firstString[n - 1]);
    String secondString = " ESP32 ";
    // Print the second string
    Serial.print("Second string = ");
    Serial.println(secondString);
    // Trim leading and trailing spaces
    secondString.trim();
    // Print the trimmed string
    Serial.print("Trimmed string = ");
    Serial.println(secondString);
    // Shorten a string
    String thirdString = "OUTPUT DATA";
    // Print the third string
    Serial.print("Third string = ");
    Serial.println(thirdString);
    String subString = thirdString.substring(0, 3);
    Serial.print("First three characters = ");
    Serial.println(subString);
    while (1) // Suspend execution
        ;
}
```

These appear between the `<style>` and `</style>` tags. Between the `<body>` and `</body>` tags you will find the three lines of text that will actually be displayed on the client device. This text inherits the properties (font, colour, margins and so on) that we've previously defined between the `<style>` and `</style>` tags.

Fig.6.5 and 6.6 respectively show the circuit and wiring layouts for the *Practical Project* – the interconnection of the DHT22/ESP32. You will find that these are identical to the diagrams that we used in our previous *Teach-In* (refer to this if you need further information on the DHT22 temperature and humidity sensor).

To test the *Practical Project* you will need to enter or download the code from the *PE* website at: <https://bit.ly/pe-downloads>

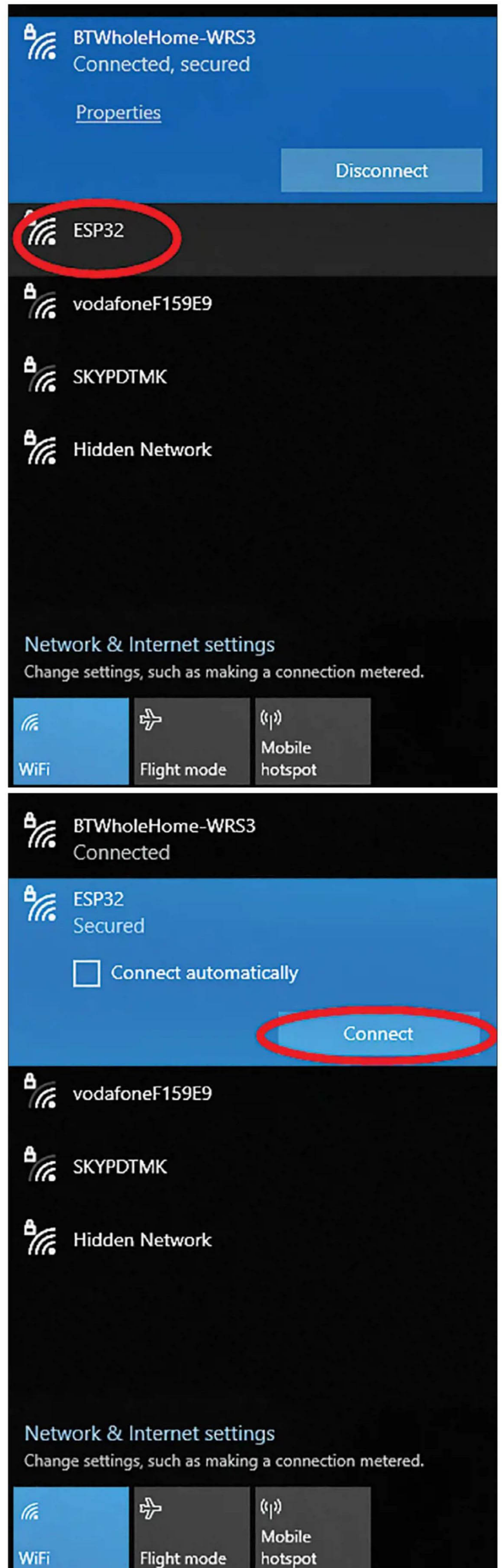
Open the file in the IDE and then compile and execute the code. If you've followed the series thus far this should be a breeze!

You might then want to check that the AP is up and running. To do this, start the Serial Monitor and then press the ESP32's reset button. You should be rewarded with a 'power-on' message like that shown in Fig.6.7. This confirms that the AP is operational.

If you look at the code in Listing 6.2 you should find that we've allocated an SSID of 'ESP32' and used '12345678' as the password. This isn't particularly imaginative, and you are welcome to change both. However, at this stage we will assume that you've kept our default values.

Fig.6.9. (Top right) Typical network selection screen on a smartphone showing the ESP32 as an available Access Point (AP).

Fig.6.10. (Bottom right) Connecting to the ESP32 AP.



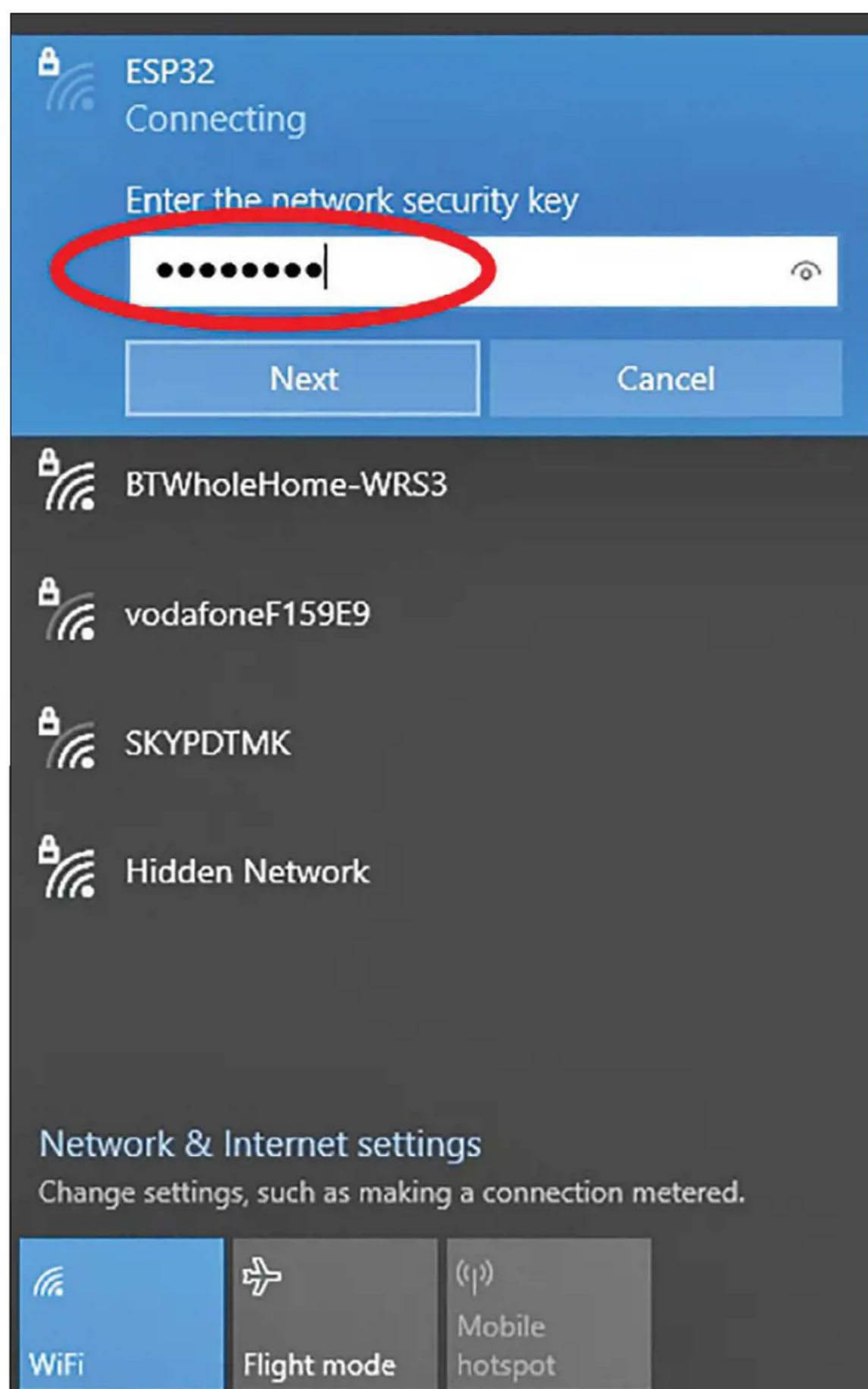
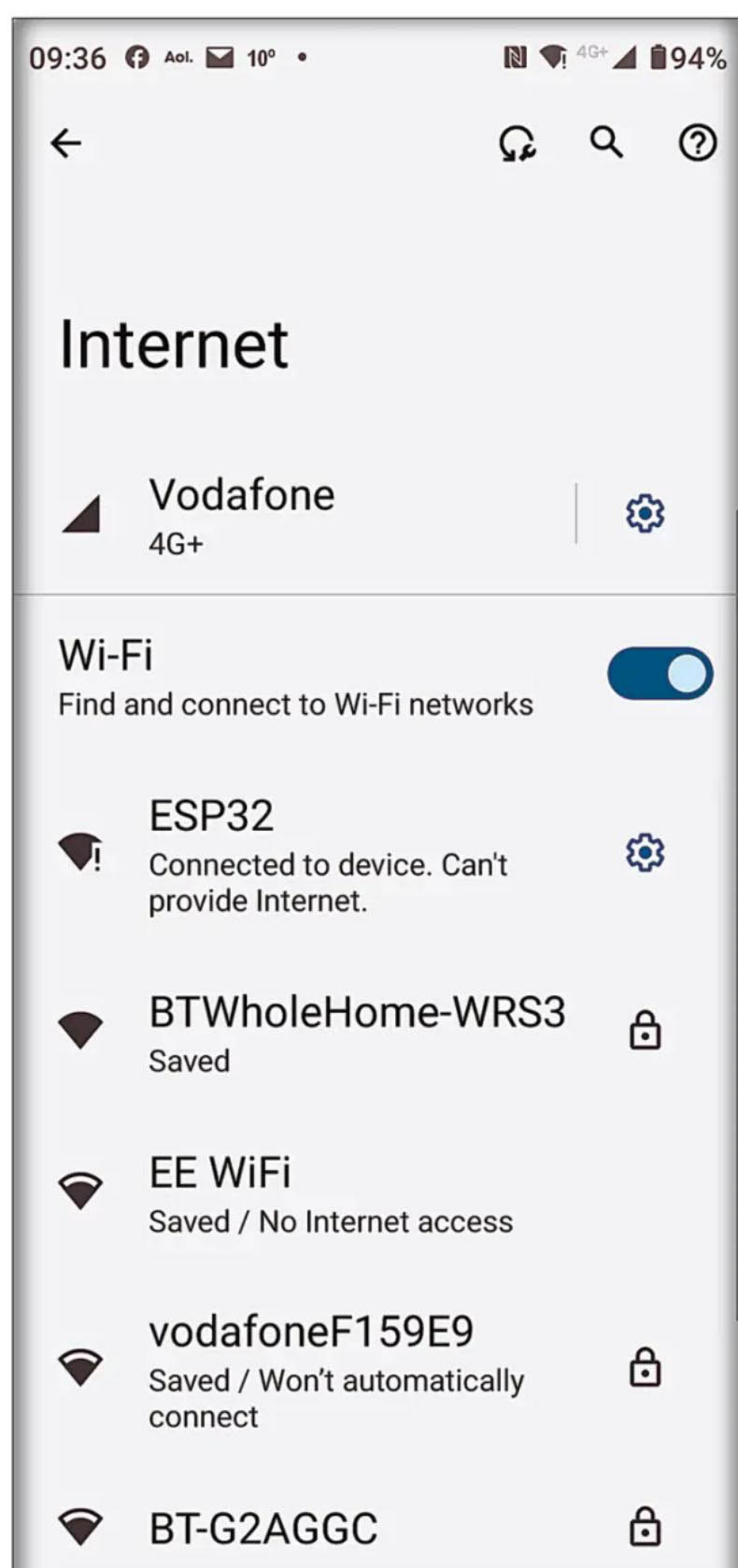


Fig.6.11. Entering the ESP32's network security key (the password).



the password '12345678' as shown in Fig.6.11.

There's one final step to complete the project. Having connected your device to the ESP32 AP you will need to open a browser and enter the IP address that we've assigned. This should be entered into the browser's address bar as '192.168.1.1'. You will be rewarded with the page that your ESP32 is currently serving, and it should look like that shown in Fig.6.12 although the displayed data will, of course, be different.

If you are using a tablet or smartphone the procedure will be similar. Fig.6.13 shows a typical list of Wi-Fi networks displayed using a smartphone while Fig.6.14 shows the page provided by the ESP32 after entering the same IP address and connecting to it.

Coding Workshop

As you begin to experiment with your own ESP32 applications you will sooner or later begin to feel the need for code that will allow you to manipulate strings of characters. This *Coding Workshop*

Fig.6.13. (Left) Selecting the ESP32 AP using an Android smartphone.

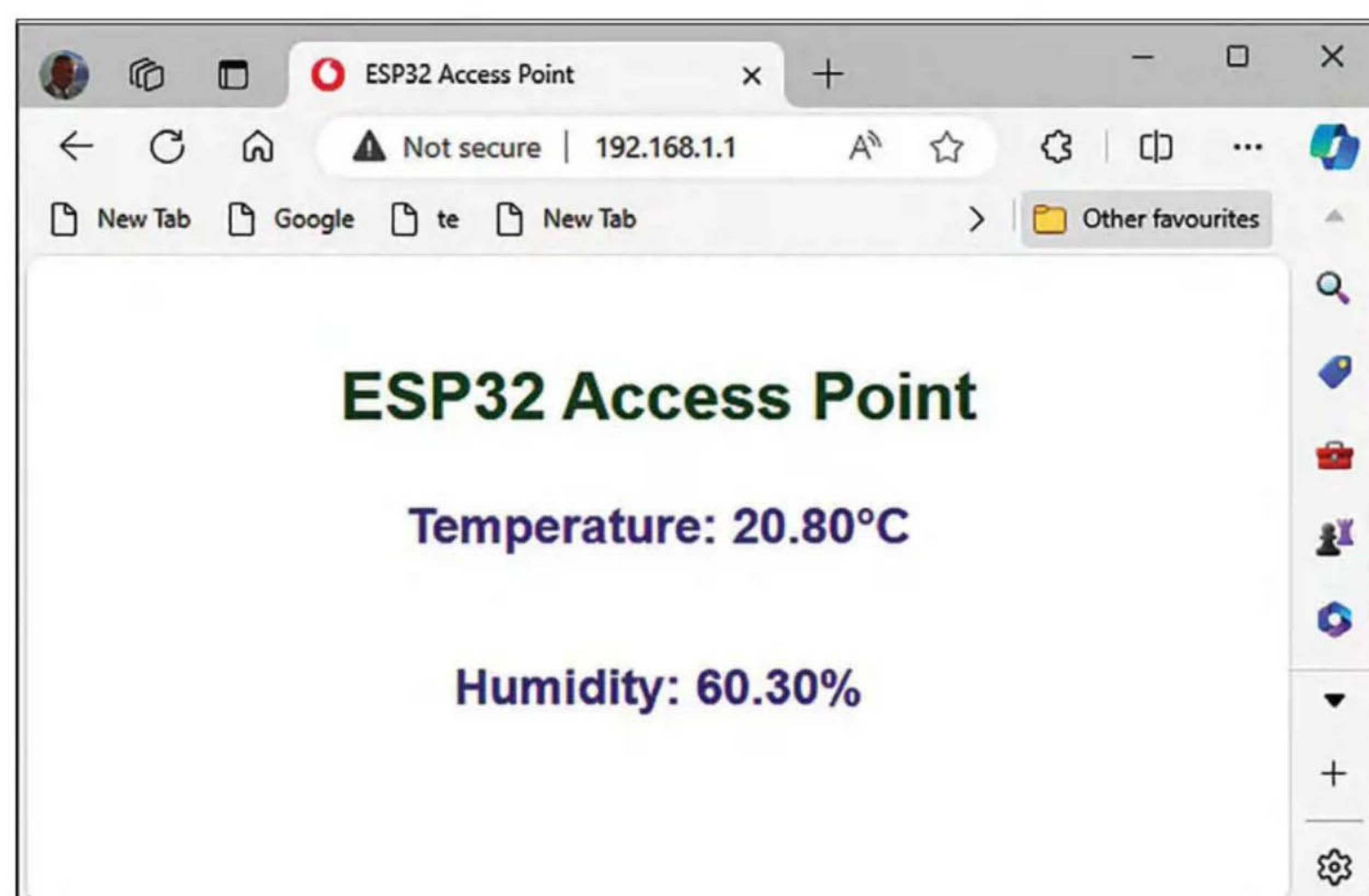


Fig.6.12. Desktop PC browser showing data served by the ESP32 in AP mode (note the IP address in the browser's address bar).

Now select your client device. This can be a desktop, laptop, tablet or smartphone equipped with Wi-Fi and a web browser. For a typical Widows PC go to Network and Internet settings, select Wi-Fi, and then click on 'Show available networks' as shown in Fig.6.8. Examine the list of available networks and then select 'ESP32' as shown in Fig.6.9. Click on 'Connect' (see Fig.6.10) and enter

Gotcha!

Arrays in the C++ language are indexed from zero. This means that the first element stored in an array has an index of 0 and for a string comprising n characters the last element will have an index of $(n-1)$

provides you with some pointers to help you along the way.

Strings are just a series of alphanumeric characters stored in array. You have already met quite a few of them, for example: HTTP started ... and text/html in Listing 6.2. Each of these strings is enclosed in quotation marks but note that an opening quote must always have a matching closing quote!

The characters in these strings are stored in a series of memory locations that we refer to as an 'array'. Here's an example of some alphanumeric text stored in an array.

```
char titleString[] = "Practical Electronics";
```

The 21 characters of 'PracticalElectronics' are stored in 21 consecutive memory locations and they are individually indexed (from 0 to 20) within the

Gotcha!

Once again, if you find that the Serial Monitor displays gobbledygook instead of a series of meaningful values, you should first check that you have set a baud rate that matches the speed that you've specified in your code. The examples in this instalment all configure the ESP32's serial port for operation at 115200 baud. To work correctly this *must* match the value that you've set in the Serial Monitor.

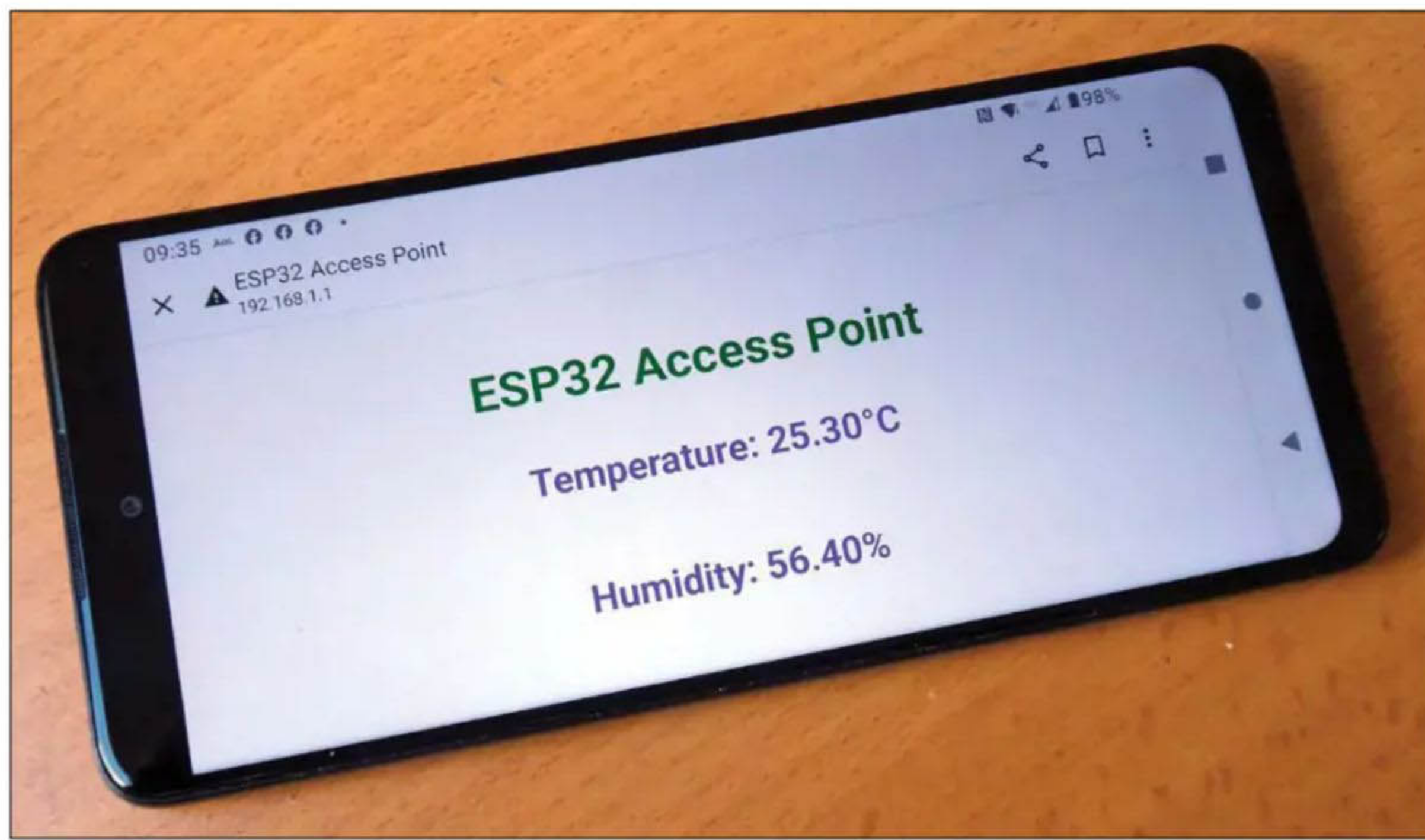


Fig.6.14. Smartphone showing data served by the ESP32 in AP mode (note the IP address in the browser's address bar).

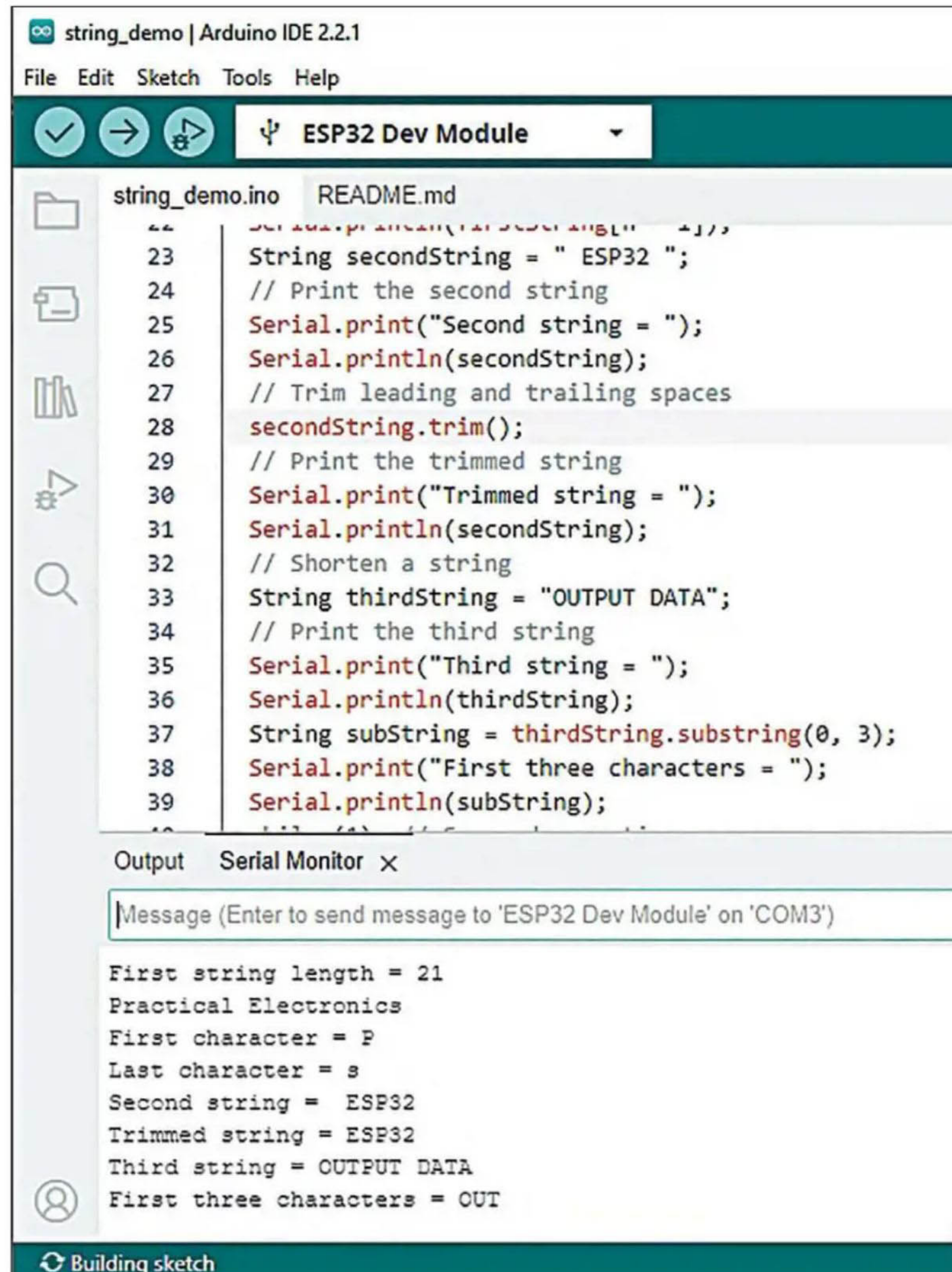


Fig.6.15. Output from Listing 6.3.

character array. For example, to extract and print the eleventh character we could use:

```
Serial.println(titleString[10]);
```

You've hopefully concluded that this would output 'E' (the *eleventh* item in the array) on the Serial Monitor.

The sample code shown in Listing 6.3 shows ways in which strings can be accessed and also modified using the `trim()` function (to remove leading and trailing spaces) and the `substring()` function (to create a substring from consecutive characters within the original string). It's worth

Gotcha!

When entering your code remember that character strings to be printed need to start and end with matching opening and closing quotation marks. An error will occur if you don't do this and the compiler will reject your code.

spending a little time experimenting with this code to get a feel for just how useful this can be. The output from Listing 6.3 is shown in Fig.6.15.

Check Point

Now that we've reached the half-way point in our *Teach-In* series *CheckPoint* replaces our usual *Teach-In Challenge*. This is a great opportunity to test your knowledge of the ESP32. Just select one answer to each of these questions. When you've finished you can check your answers at the bottom of page 16. Good luck!

1. The maximum integer number that can be represented using unsigned eight bits is:

- (a) 7
- (b) 15
- (c) 255.

2. In-line comments are preceded by:

- (a) *
- (b) //
- (c) \$.

3. In which band does ESP32 Wi-Fi normally operate?

- (a) 1.2GHz
- (b) 2.4GHz
- (c) 5GHz.

4. The expression `delay(50);` will suspend the execution of code for:

- (a) 5s
- (b) 50s
- (c) 0.050s.

5. Analogue output from an ESP32 is based on:

- (a) Amplitude modulation (AM)
- (b) Frequency modulation (FM)
- (c) Pulse-width modulation (PWM).

6. 16-bit numbers with no decimal points are referred to as:

- (a) Bytes
- (b) Integers
- (c) Floating point.

7. In the first byte of an I²C transfer, how many bits are used to convey the address?

- (a) Four bits
- (b) Seven bits
- (c) All eight bits.

8. When an LED is conducting, its:

- (a) Anode is positive with respect to its cathode
- (b) Cathode is positive with respect to its anode
- (c) Cathode is at the same potential as its anode.

9. To set the state of a digital I/O pin you would use:

- (a) `outputSet();`
- (b) `digitalWrite();`
- (c) `outputWrite();`

10. The I²C interface standard allows for a maximum of:

- (a) 15 devices
- (b) 63 devices
- (c) 127 devices.

11. The speed at which serial data is transferred is often quoted in:

- (a) Baud
- (b) Hertz
- (c) Mbyte.

12. In an I²C bus:

- (a) Only slaves can be present
- (b) Only one bus master can be present
- (c) More than one bus master can be present.

13. The ESP32's magnetic sensor is based on:

- (a) Hall effect
- (b) Ohm's Law
- (c) Charge conservation.

14. Which one of the following gives the typical forward current for a small LED indicator?

- (a) 5mA
- (b) 50mA
- (c) 500mA

15. Internal pull-up and pull-down resistors are provided for use with:

- (a) All of the ESP32's digital I/O pins
- (b) None of the ESP32's digital I/O pins
- (c) Some of the ESP32's digital I/O pins.

16. On an SPI bus the bus lines are:

- (a) Usually pulled high and driven low when data is present
- (b) Usually pulled low and driven high when data is present
- (c) Always left floating regardless of whether data is present.

17. When $x = 2$ and $y = 3$ the maths expression `pow(x, y)` will evaluate to:

- (a) 6
- (b) 8
- (c) 9.

18. The maximum HIGH state voltage that can appear on any of the ESP32's GPIO pins is:

- (a) +2V
- (b) +3.3V
- (c) +5V.

19. Which one of the following lines of code will generate an error?

- (a) `int a = 3.142;`
- (b) `delay(60000);`
- (c) `pinMode(19, OUTPUT);`

20. Switch bounce results in:

- (a) A rapid and undesirable change in logic levels
- (b) The inability of a switch to remain in the currently selected state
- (c) Unwanted delay before the contacts of a switch open or close.

Answers are at the bottom of page 16.

How did you do?

If you scored 18 or more, you've already become an ESP32 expert. If you scored between 15 and 18, you're already well on the way. If you scored less than 15 it's worth taking another look at the earlier parts of this *Teach-In* series.

Next month

In next month's *Teach-In* we will be introducing Network Time Protocol (NTP). Our *Practical Project* will take the form of an accurate clock that displays network time on a 16×2 character LCD. *Coding Workshop* will look at arrays and array handling.

**CRICKLEWOOD
ELECTRONICS**
Established 1981

Your best bet since MAPLIN
Chock-a-Block with Stock

Visit: www.cricklewoodelectronics.com

Or phone our friendly knowledgeable staff on **020 8452 0161**

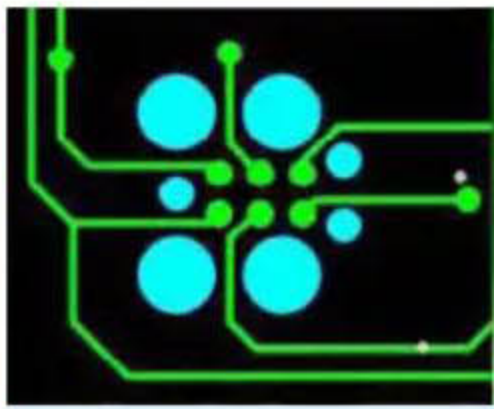
Components • Audio • Video • Connectors • Cables
Arduino • Test Equipment etc, etc

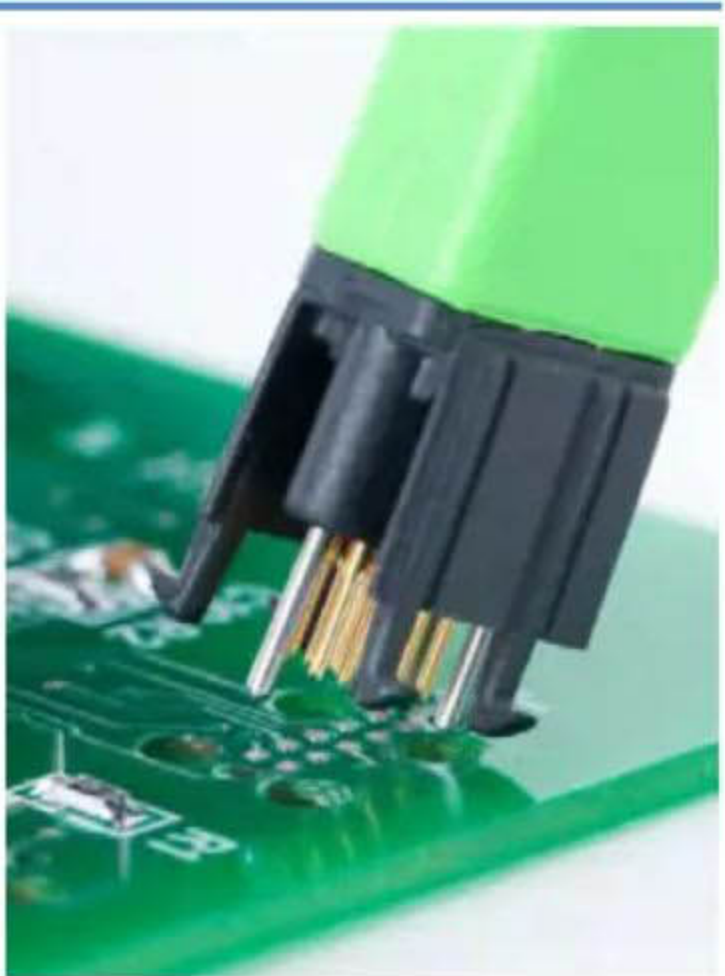
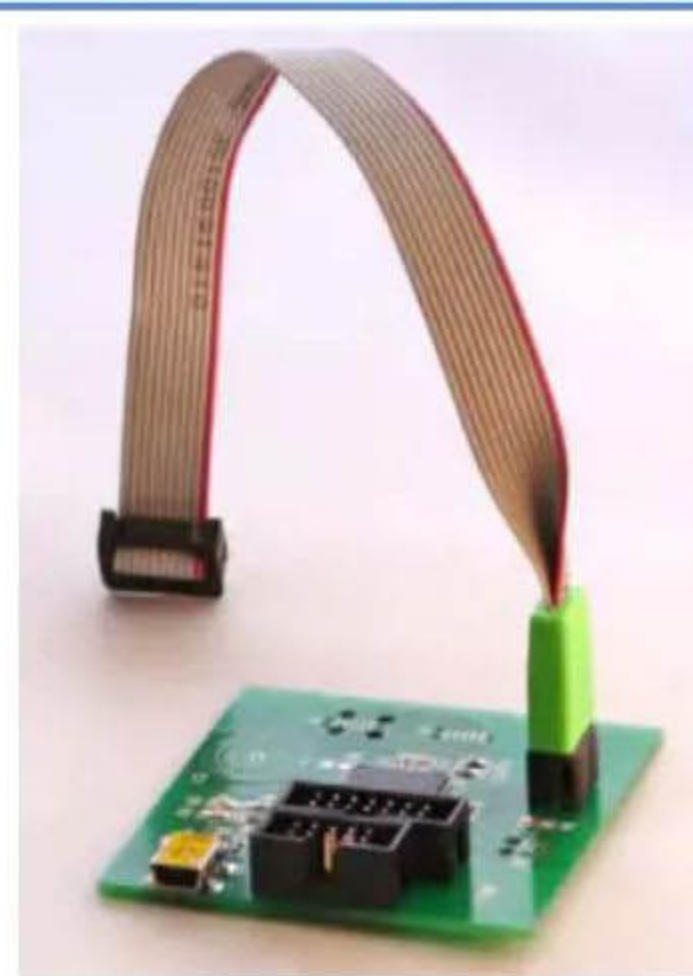


Visit our Shop, Call or Buy online at:
www.cricklewoodelectronics.com

020 8452 0161

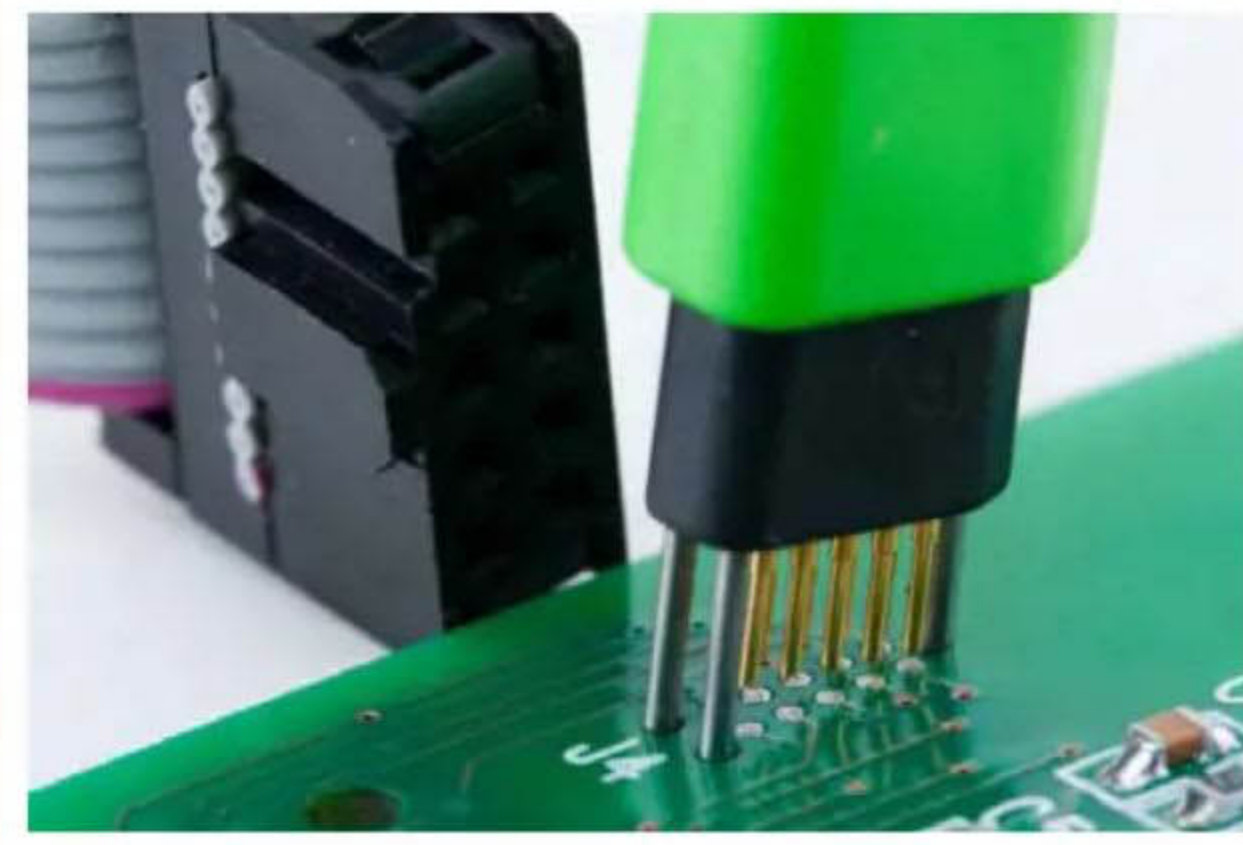
Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

**Tag-Connect**



JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!



Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC • dsPIC • ARM • MSP430 • Atmel • Generic JTAG • Altera Xilinx • BDM • C2000 • SPY-BI-WIRE • SPI / IIC • Altium Mini-HDMI • & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

CLASSIFIED ADVERTISING



Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2.5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to 'Practical Electronics'. **VAT must be added.** Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 07973518682 Email: pe@electronpublishing.com

For rates and further information on display and classified advertising please contact our Advertisement Manager, Matt Pulzer – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD

Suppliers of Electronic Components

www.bowood-electronics.co.uk

Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222

Send large letter stamp for Catalogue

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS?

For free stock list and/or advice, please contact me: geoffdavies337@gmail.com
Telephone: 01788 574774

PIC DEVELOPMENT KITS, DTMF kits

and modules, CTCSS Encoder and Decoder/Display kits.

Visit www.cstech.co.uk

Electrical Industries Charity (EIC)

We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.

www.electricalcharity.org



COMPONENTS AS JOB LOT

Large stock. Transistors, Diodes, Logic, Memory and Analogue parts. 1000s of passive items. Catalogued and sorted. 'New Old Stock' in racks.
Email: rojobarn@gmail.com



COAST ELECTRONICS

BREAKOUTS-COMPONENTS-
CONTRACT DESIGN-3D PRINTER PARTS-
MUSICAL-MICROCONTROLLERS

WWW.COASTELECTRONICS.CO.UK

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO

Circuits Electric Machinery Mechatronics

Web: www.akennypatentm.com

Email: Enquiries@akennypatentm.com

Tel: 0789 606 9725

ADVERTISING INDEX

CRICKLEWOOD ELECTRONICS	68
ESR ELECTRONIC COMPONENTS	50
HAMMOND ELECTRONICS Ltd	11
JPG ELECTRONICS	72
FLOWCODE	15
MICROCHIP	Cover (ii)
PEAK ELECTRONIC DESIGN	Cover (iv)
POLABS D.O.O.	32
QUASAR ELECTRONICS	3
SILICON CHIP	5
STEWART OF READING	59
TAG-CONNECT	68
TERRINGTON COMPONENTS	13

Advertisement offices

Matt Pulzer

Electron Publishing Ltd
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 07973 518682

Email pe@electronpublishing.com

Web www.electronpublishing.com

For editorial contact details see page 7.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
AUGUST 2024		
Arduino ESR Meter combined L/C-ESR.....	04106182	9.99
Arduino ESR Meter separate ESR	04106181	7.99
Watering System Controller	15110231	11.99
JULY 2024		
Dynamic NFC Tag (22 x 31mm)	06101231	5.95
Dynamic NFC Tag (86 x 54mm)	06101232	7.95
Reciprocal Frequency Counter	CSE230101C	8.95
Reciprocal Frequency Counter, lid 1mm thick black	CSE230102	7.95
Pi Pico-based Thermal Camera.....	04105231	6.95
JUNE 2024		
Loudspeaker Test Jig.....	04106231	12.95
MAY 2024		
GPS-Disciplined Oscillator	04103231	12.95
Dual RF Amplifier.....	CSE220602A	5.95
Songbird	08103231	7.95
APRIL 2024		
Basic RF Signal Generator	CSE221001	8.99
Basic RF Signal Generator (black front panel).....	CSE220902B	9.99
Silicon Chirp Cricket	08101231	7.99
Test Bench 'Swiss Army Knife'	04110221	9.99
MARCH 2024		
Digital Volume Control Potentiometer SMD version	01101231	8.95
Digital Volume Control Potentiometer through-hole version.....	01101232	9.95
Advanced SMD Test Tweezers body	04106221	6.95
Advanced SMD Test Tweezers legs (pair).....	04106212	7.95
FEBRUARY 2024		
Active Mains Soft Starter.....	10110221	10.95
JANUARY 2024		
Q Meter	CSE220701	8.95
Q Meter (black solder mask)	CSE220704	7.95
Raspberry Pi Pico W BackPack.....	07101221	8.95
DECEMBER 2023		
Digital Boost Regulator.....	24110224	8.95
Dual-Channel Power Supply for Breadboards	04112221	11.95
Display Adaptor for the Breadboard PSU	04112222	9.95
NOVEMBER 2023		
LC Meter Mk3	CSE220503C	9.95
DC Supply Filter for vehicles.....	08108221	8.95
Discrete Audio Op Amp PCB	AO1-JUL23	9.95
Discrete Audio Op Amp PCB + essential components	AO2-JUL23	17.95
OCTOBER 2023		
Buck/Boost Charger Adaptor	14108221	9.95
PIC Breakout Board for SOIC parts.....	24110225	5.95
PIC Breakout Board for DIP parts.....	24110222	5.95
AVR64DD32 Breakout board.....	24110223	5.95
Automatic Train Controller.....	09109221	6.95
Chuff Sound module.....	09109222	6.95
SEPTEMBER 2023		
Mini LED Driver.....	16106221	7.95
New GPS-Synchronised Clock.....	19109221	9.95
Wide-Range Ohmmeter	04109221	14.95

PROJECT	CODE	PRICE
AUGUST 2023		
110dB RF Attenuator	CSE211003	9.95
Universal Battery Charge Controller (2023 update).....	14107192	9.95
Wide-Range OhmMeter	04109221	12.95
JULY 2023		
Multimeter Checker/Calibrator	04107221	10.95
MIDI Spectral Sound Synthesiser (full kit – see p.25).....	N/A	N/A
JUNE 2023		
Arduino Programmable Load.....	04105221	8.95
Buck-Boost LED Driver	16103221	8.95
MAY 2023		
Precision AM-FM DDS Signal Generator	CSE211002	14.95
Improved SMD Test Tweezers programmed PIC.....	0410621PIC	12.95
AO 2x-dual-to-quad through-hole (pack of five)	AO1-MAY23	6.95
AO 2x-dual-to-quad SMD (pack of five).....	AO2-MAY23	6.95
AO 2x-single-to-dual through-hole (pack of five).....	AO3-MAY23	6.95
AO 2x-single-to-dual SMD (pack of five)	AO4-MAY23	6.95
APRIL 2023		
500W Amplifier Module	see p.22, April 2023	
Clipping Indicator (per channel)	01112211	7.95
CD Welder Power Supply (one needed)	29103221	9.95
CD Welder Controller (one needed)	29103222	9.95
CD Welder Energy Storage module (several needed)	29103223	7.95
AO Universal Dual Op Amp Board	AO1-APR23	9.95
AO Stereo RIAA precision passives kit for dual op amp	AO2-APR23	8.95
MARCH 2023		
Pico BackPack.....	07101221	9.95
Semaphore Signal (controller)	09103221	7.95
Semaphore Signal (blade)	09103222	5.95
FEBRUARY 2023		
Tesla Coil driver board.....	26102221	9.95
Tesla Coil potentiometer board	26102222	5.95
Cooling Fan Controller & Loudspeaker Protector	01102221	8.95
Remote Gate Controller	11009121	12.95
JANUARY 2023		
Classic LED Metronome – 8-LED.....	23111211	7.95
Classic LED Metronome – 10-LED.....	23111212	8.95
Multi-Channel Speaker Protector – 6 channel	01101221	9.95
Multi-Channel Speaker Protector – 4 channel	01101222	7.95
Remote Control Range Extender – IR-to-UHF	15109212	5.95
Remote Control Range Extender – UHF-to-IR	15109211	7.95
AO Universal Single Op Amp Board.....	AO1-JAN23	7.95
DECEMBER 2022		
Hummingbird Amplifier	01111211	9.95
SMD Trainer PCB	29106211A	8.95
SMD Trainer PCB + parts	29106211B	13.95
NOVEMBER 2022		
USB Cable tester – main PCB.....	04108211	12.95
USB Cable tester – front panel	04108212	5.95
USB Cable tester – optional panel.....	SC5970	5.95
Model Railway Carriage Lights – PCB	09109211	6.95
AO transfrm PCB – standard VTX-A range	VTX-101-007	6.95
AO transfrm PCB – dual-outline VTX102-3001/101-3002 ...	VTX-Dual	6.95
OCTOBER 2022		
SMD Test Tweezers – PCB and pair of tweezer arms.....	04106211-2	11.95
SMD Test Tweezers – programmed PIC12F1572-I/SN.....	0410621A	7.95
Tele-com	12110211	12.95

PCBs for most recent *PE/EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Next Month – in the September issue

Build Your Own Calibrated Microphones

This phantom-powered, balanced and calibrated microphone is ideal for measurement applications but is also great for recording work. You can choose from multiple different microphone capsules that can be calibrated to produce an almost flat response.

Modern PIC Programming Adaptor

Program most newer PIC microcontrollers out-of-circuit with this PIC Programming Adaptor. It works using a PICKit or SNAP programmer and suits 8- to 40-pin micros. With external adaptors, you can program SMD chips in SOIC, SSOP or TSSOP packages.

Salad Bowl Speakers

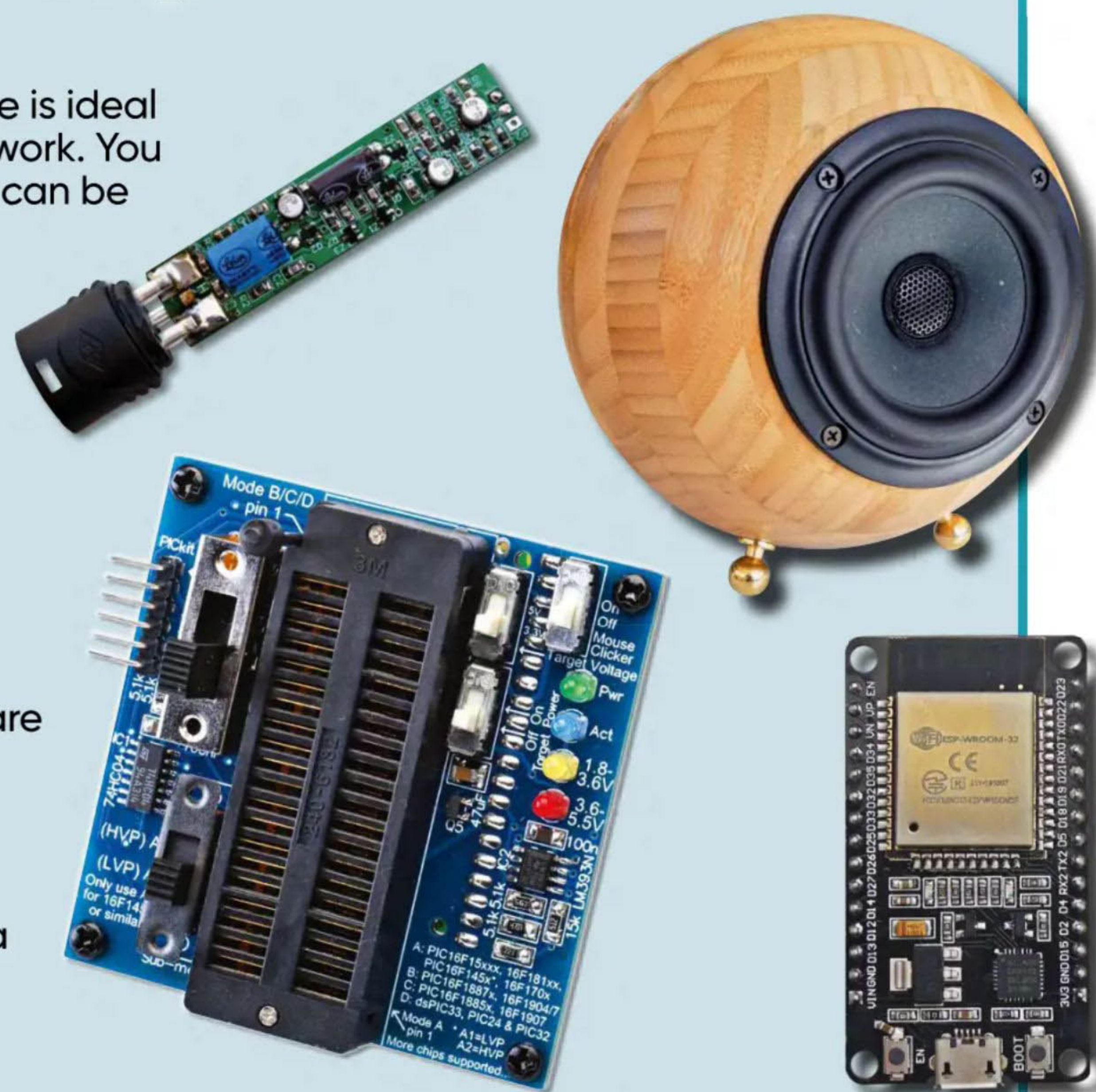
These stylish speakers are surprisingly good performers considering they're made using salad bowls from IKEA. They are simple to build but they can handle 50W RMS per channel.

Teach-In

Next month, we will introduce Network Time Protocol (NTP), and create an accurate clock that displays network time on a 16 × 2 character LCD.

PLUS!

All your favourite regular columns from *Cool Beans* and *Circuit Surgery*, to *Audio Out*, *Fox Report*, *Techno Talk* and *Net Work*.



On sale 1 August 2024

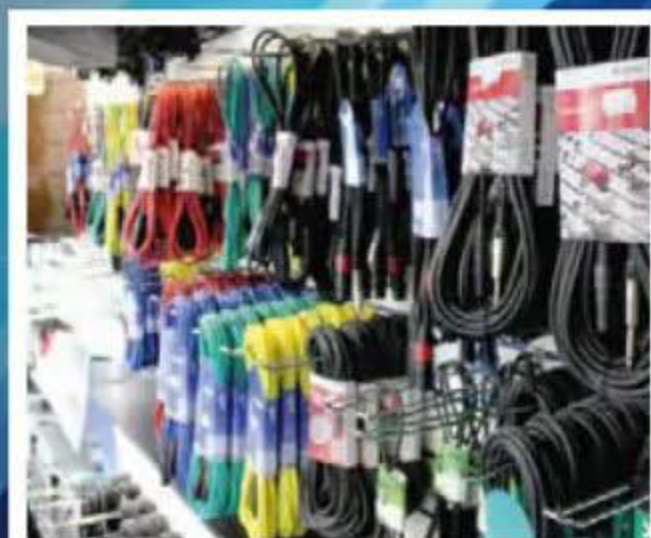
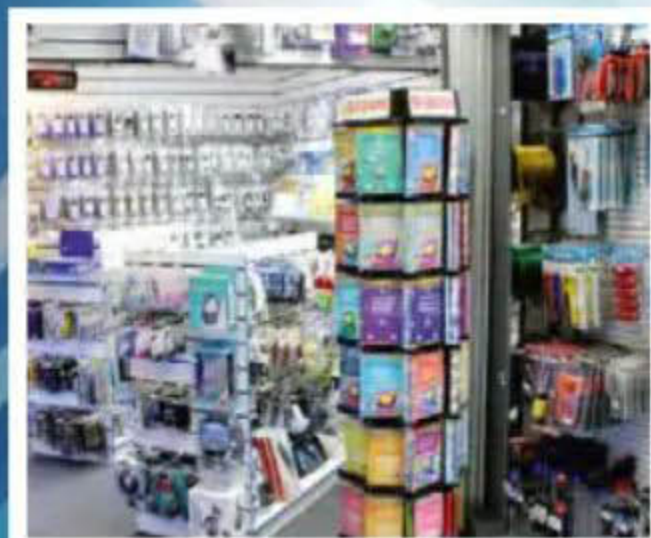
Content may be subject to change

WELCOME TO JPG ELECTRONICS

Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
 - ICs, Project Boxes, Relays & Resistors
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more...



T: 01246 211 202

E: sales@jpgelectronics.com

JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB

W: www.jpgelectronics.com



Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

NEW subscriptions hotline!

Practical Electronics

We have changed the way we sell and renew subscriptions. We now use 'Select Publisher Services' for all print subscriptions – to start a new subscription or renew an existing one you have three choices:

1. Call our NEW print subscription hotline:
01202 087631, or email: pesubs@selecttps.com
2. Visit our shop at: www.electronpublishing.com
3. Send a cheque (payable to: 'Practical Electronics') with your details to:
Practical Electronics Subscriptions, PO Box 6337,
Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £33.99 (6 months); £59.99 (12 months); £114.99 (2 years). EUROPE: airmail service, £38.99 (6 months); £69.99 (12 months); £129.99 (2 years). REST OF THE WORLD: airmail service, £46.99 (6 months); £84.99 (12 months); £164.99 (2 years). Payments payable to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom. Email: pesubs@selecttps.com. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics

The UK's premier electronics and computing maker magazine

Circuit Surgery

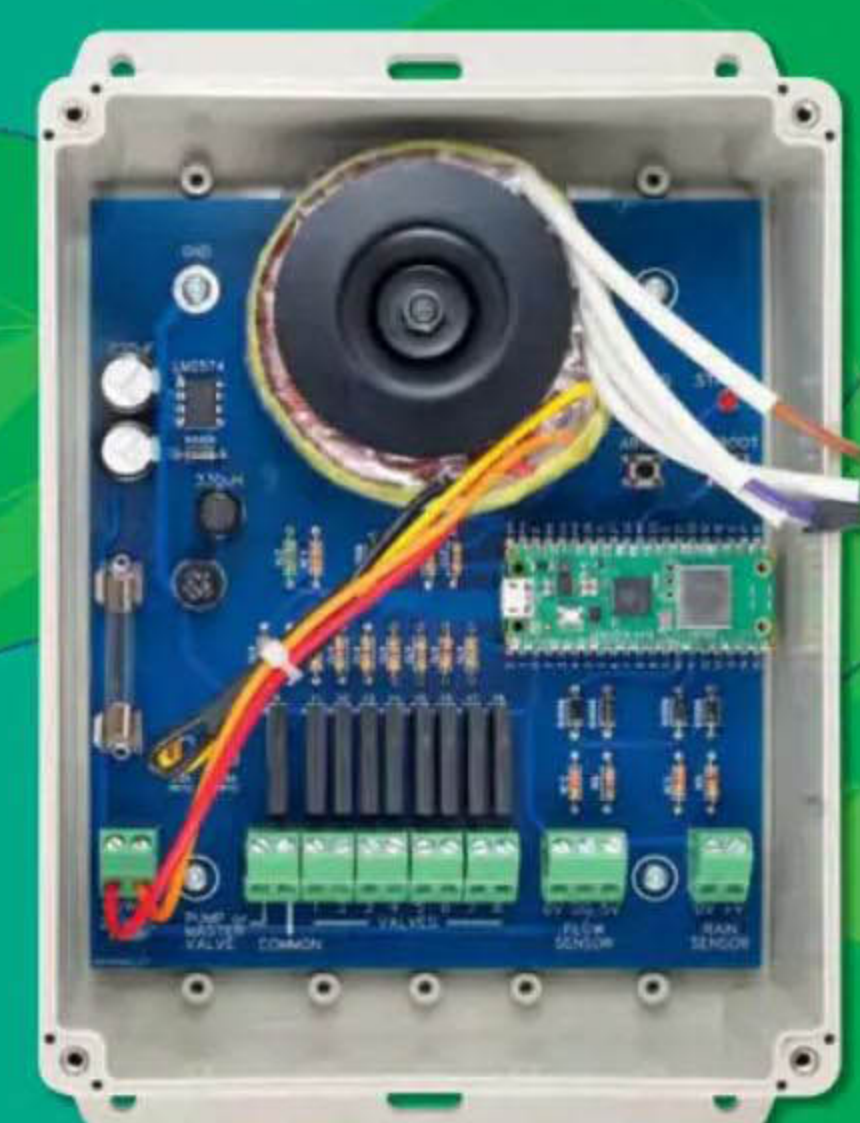
Topics in digital signal processing – the frequency domain

Teach-In 2024

Create a Wi-Fi Access Point with the ESP32 SoC microcontroller

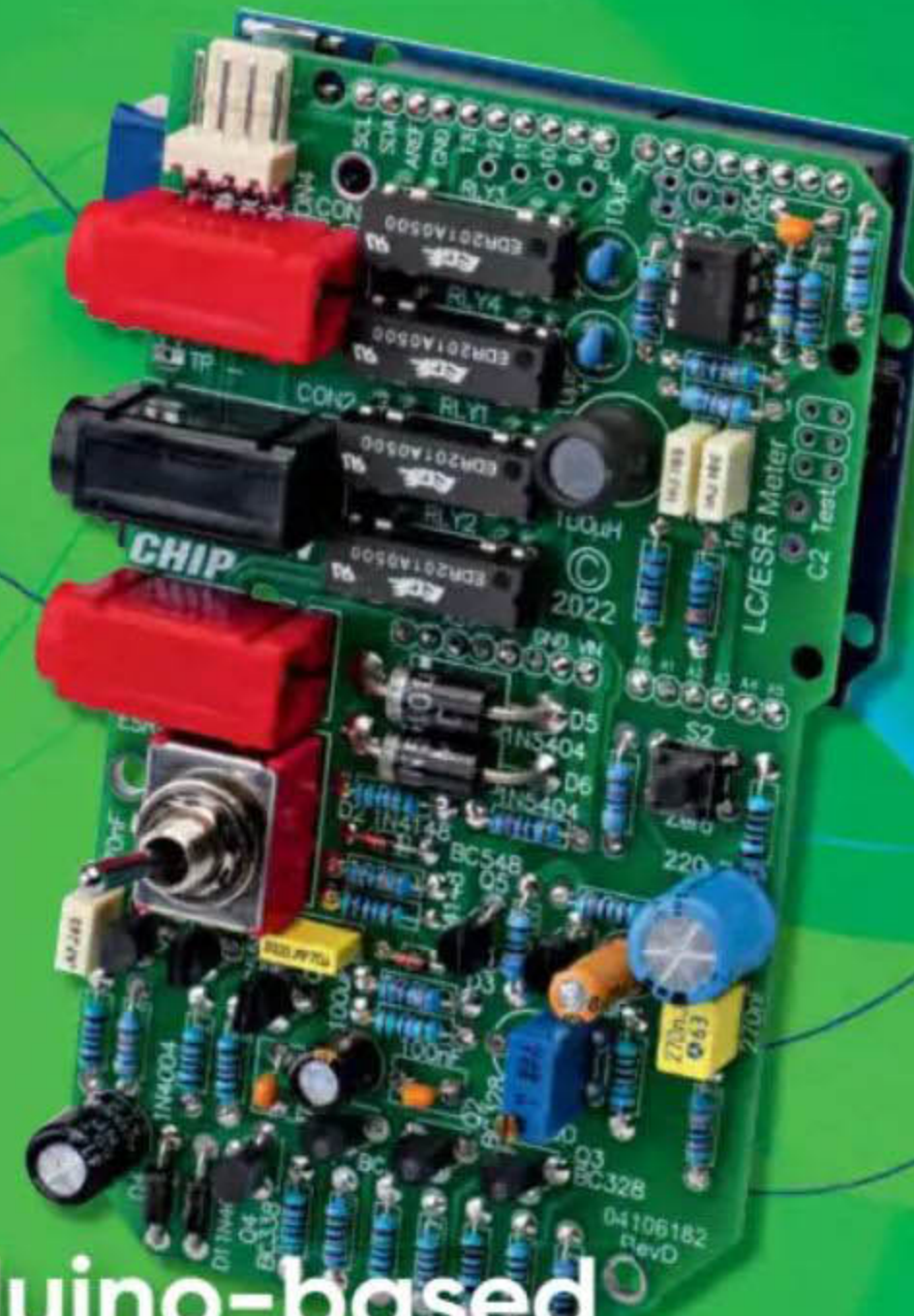
Audio Out

Mechanical switches in audio



WebMite Watering System Controller

WIN!
Microchip
PolarFire SoC
Discovery Kit
WIN!



Arduino-based L/C+ESR Meter



Teach-In 2024
Learn to use the ESP32

WebMite: PicoMite meets the Web

PLUS!

Techno Talk – That makes so much sense!

Cool Beans – DIY logic probes

Net Work – Passkeys and online security

Aug 2024 £5.99



www.electronpublishing.com |  @practicalelec |  practicalelectronics

Practical Electronics

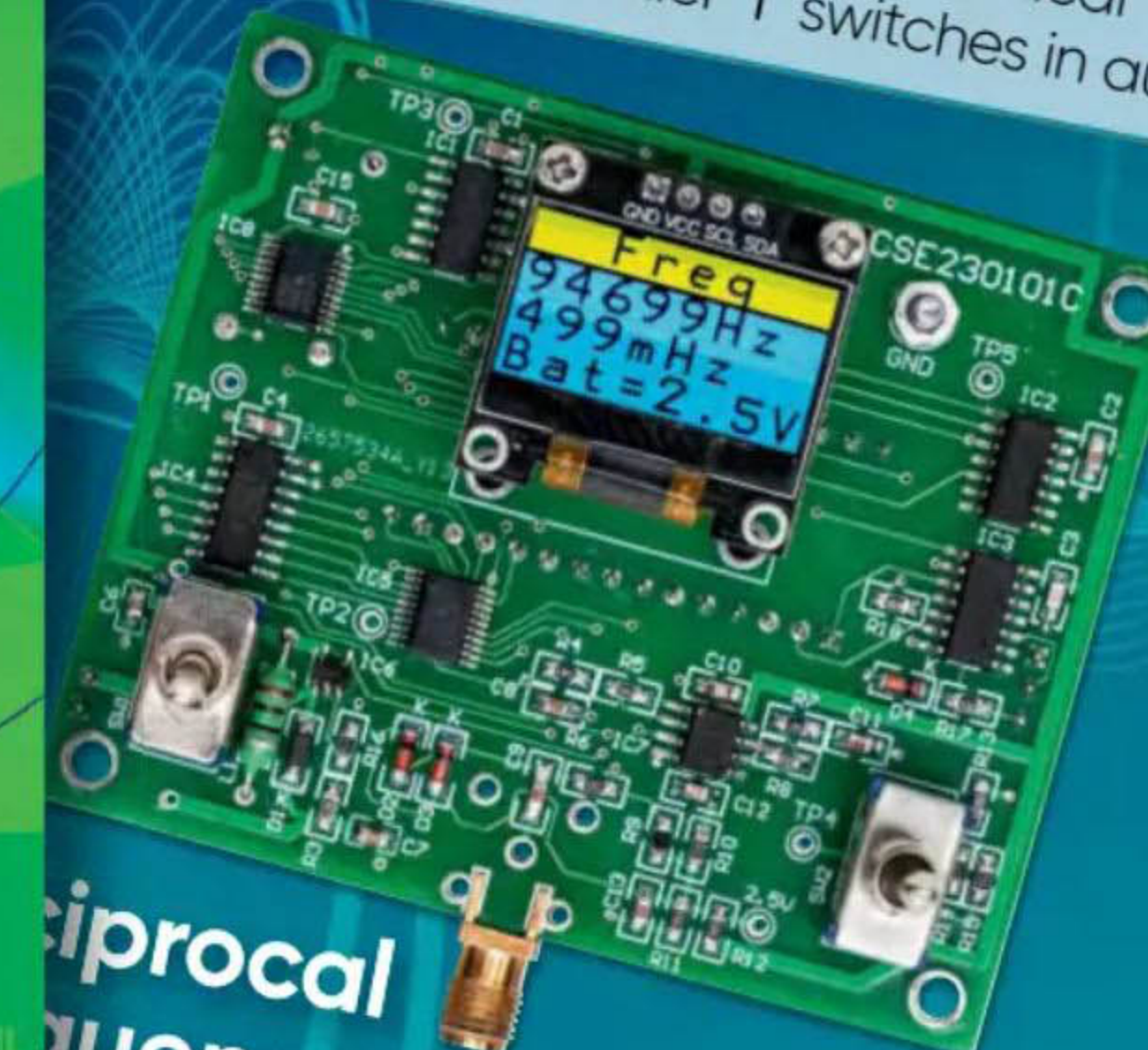
The UK's premier electronics and computing maker magazine

Teach-In 2024

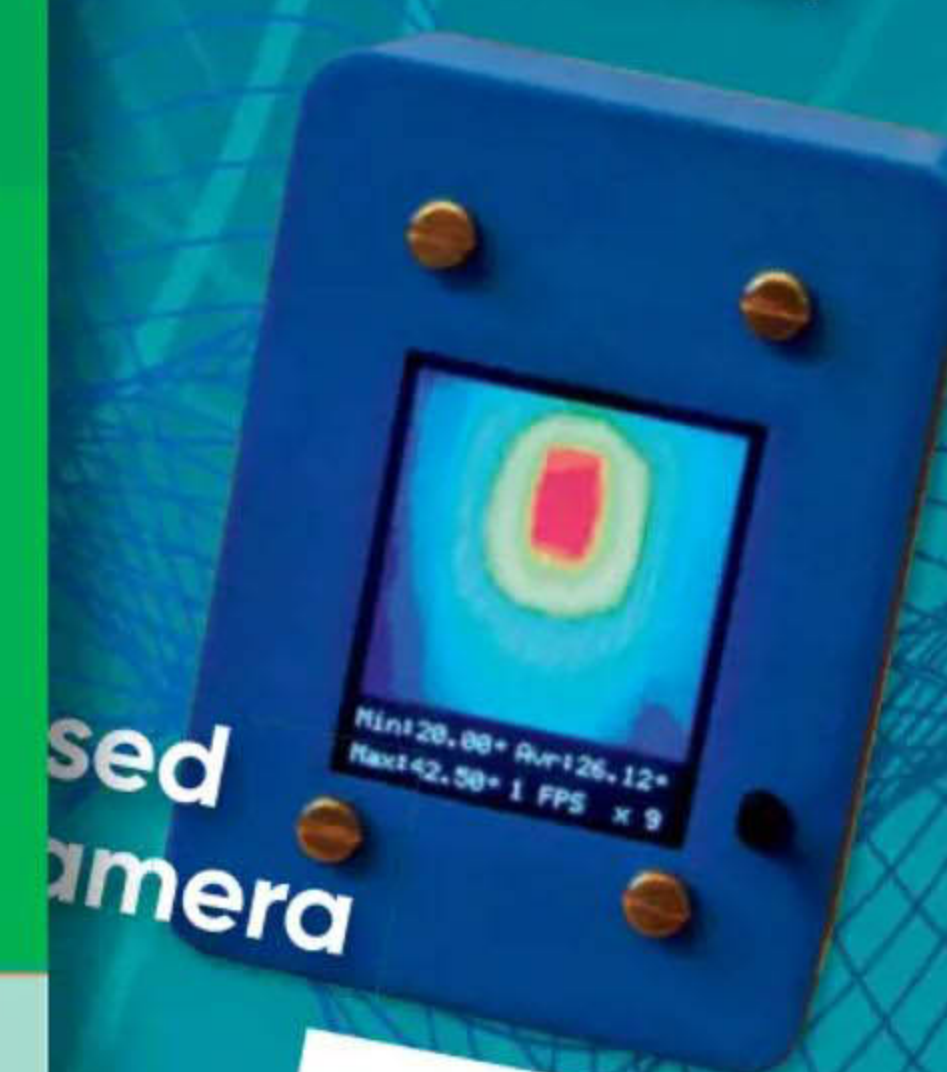
Creating the environment with the ESP32 SoC microcontroller

Audio Out

Mechanical switches in audio



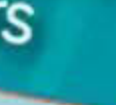
Reciprocal Frequency Counter



Used camera

Jul 2024 £5.99



www.electronpublishing.com |  @practicalelec |  practicalelectronics

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

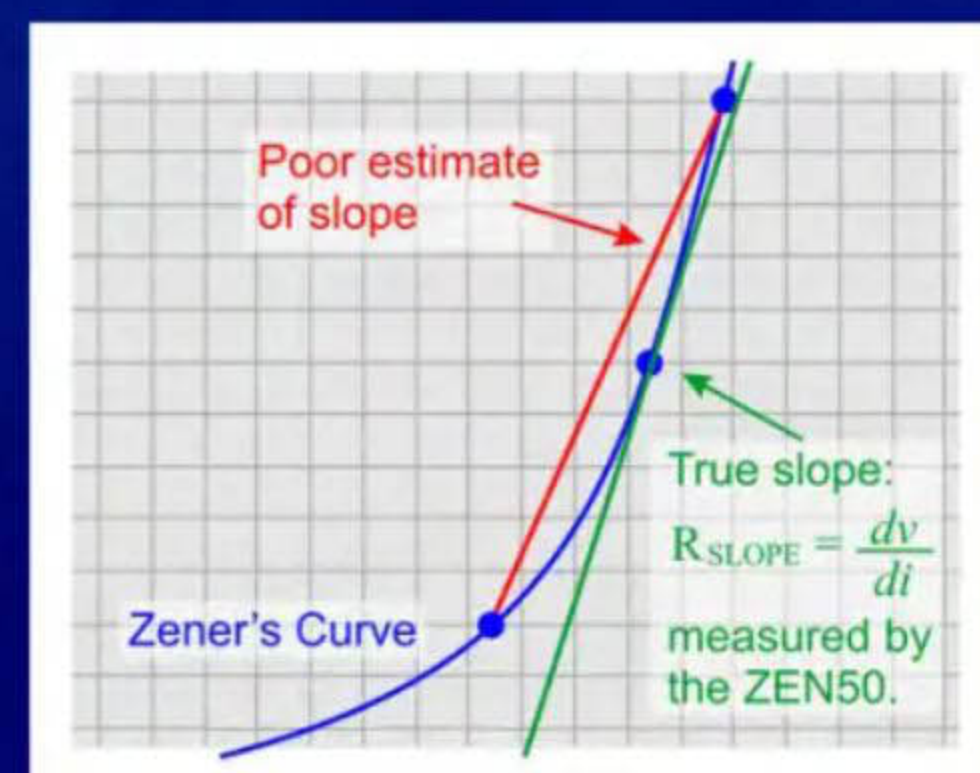
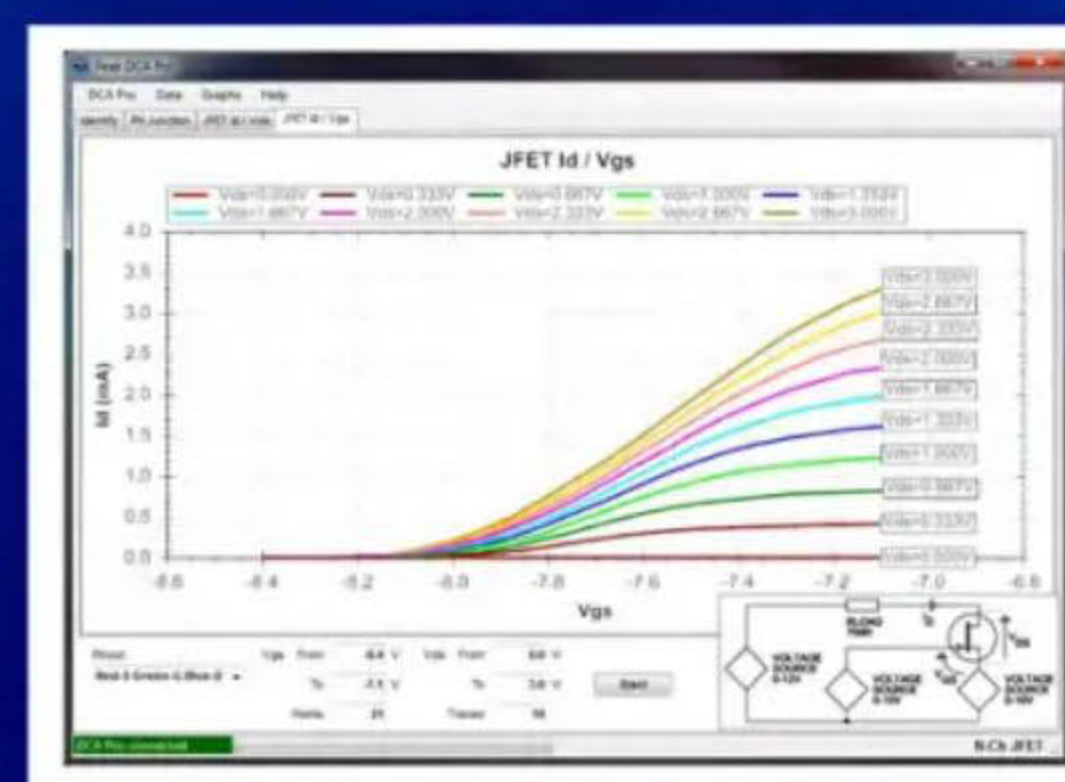
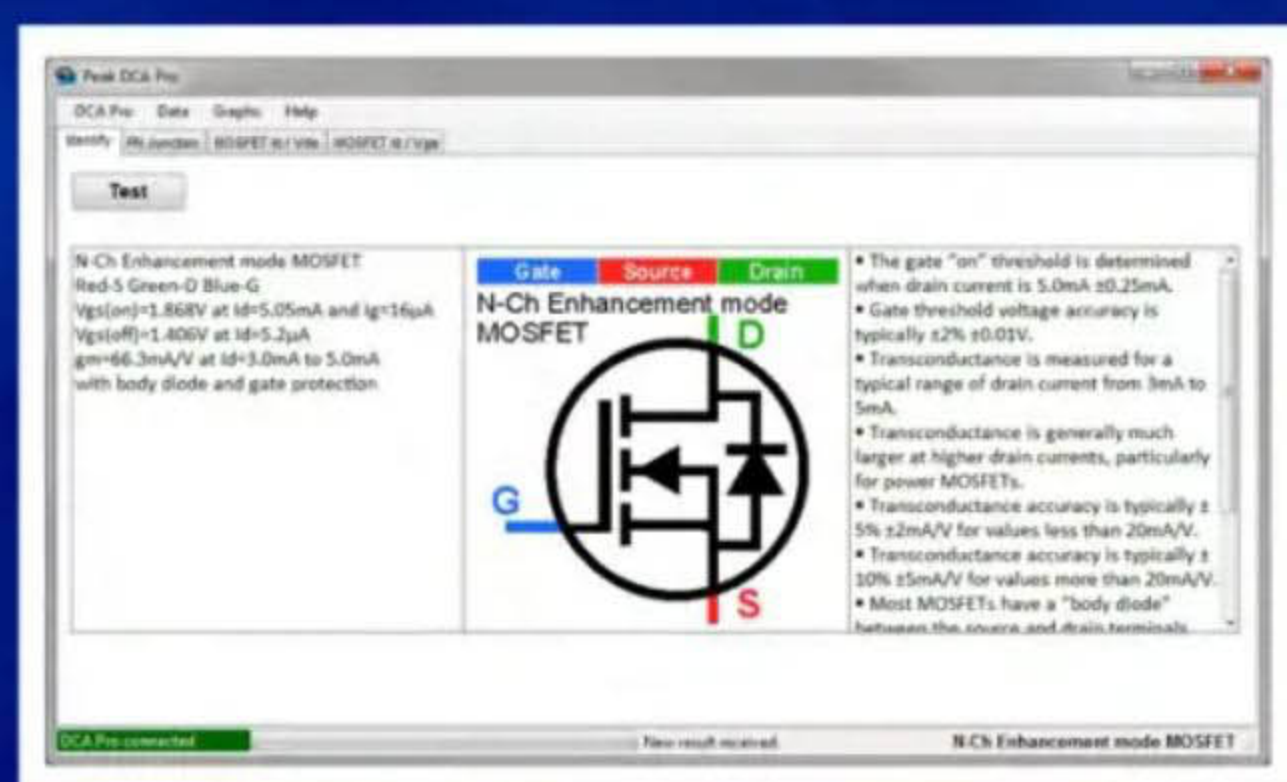
www.electronpublishing.com

Use the code **PE10**
to claim your **10% discount!**
(Offer to end soon)

Just imagine if you could:

Test almost any component
Identify unmarked/unknown components
Identify the pinout of components
Measure the parameters of components
Match components
Wind your own inductors
Fine tune your trimmer capacitors
Identify ideal operating points
Plot full datasheet-style charts
Identify faulty or degraded components
Identify alternatives

With Peak, YOU CAN.



Get guidance from the actual designers and manufacturers
here in the heart of Derbyshire, UK.



Tel. 01298 70012 for genuine advice
order: www.peakelec.co.uk